ATOM NIEUWS

JAARGANG : 13
NUMMER : 4

STOP NOG EVEN

ROLAND !





MET DIE "3e GENERATIE-ATOM"!!



FEDERATIE VAN ATOMCLUBS NEDERLAND - BELGIE.

Voorzitter :	Secretaris:	Penningmeester:
P.v.Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JP Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
tel. 02997-1902	tel. 08373-13757	tel.03461-3495

Contributie 1994 : fl. 25,00 : Atom Computerclub : Giro 5244293.

Redactie Atom Nieuws	Redactieadres A.N.	Ledenadministratie
B.Tossaint 043-431675	B.Tossaint	T.Rutten
W.Truijen 00-3289564792	Fatimaplein 85	Berkenlaan 24
R.Leurs 046-370650	6214 TW Maastricht	3737 RN Groenekan
	tel. 043-431675	tel. 03461-3495

UITERSTE DATUM INLEVERING KOPY VOOR NR. 14-1 : 1 MRT 1995

Clubwinkel	ATOM-BULLETIN-BORD speciaal v. ATOM-in-PC	
J.Hartog	R.Bronsdijk	ij Unicorn BBS (H.Derksen)
Keyenbergseweg 60		ij 085-425506 in de gebieden
6871 WK Renkum	inloggen op	ij ATOMFILE en ATOMMESSAGE de
tel. 08373-13757	tel.020-6512861	ij laatste versies ATOM-PC - systeemsoftware .

De Clubwinkel :

80-koloms-video-kaart excl. onderdelen	fl. 5,00
Combikaart 91 versie 1 : zie SPS-Printservice	
Z-80-kaart voor CP/M , exlusief onderdelen	fl. 10,00
ACORN NIEUWS 1982, 97 pagina's samenvatting	fl. 1.00
ATOM NIEUWS jaargang 1983 , +/- 450 pag.	fl. 1.00
ATOM NIEUWS Jaargang 1984 , +/- 650 pag.	fl. 1,00
ATOM NIEUWS Jaargang 1985 , +/- 650 pag.	fl. 1,00
ATOM NIEUWS Jaargang 1986 , +/- 500 pag.	fl. 1,00
ATOM NIEUWS Jaargang 1987 , +/- 300 pag.	fl. 1,00
ATOM-WARE : deel 1 : Atom-basic interpreter , 98 pag.	fl. 1,00
ATOM-WARE : deel 2 : Atom-disk operating syst.68 pag.	fl. 1,00
ATOM-WARE : deel 3 : Monitor operating system 80 pag.	fl. 1,00

Levering geschiedt via uw regionale penningmeester, of rechtstreeks, via de penningmeester van de federatie . Bij rechtstreekse bestelling stort U het bedrag van het gewenste artikel , vermeerderd met fl. 4,00 portokosten , op de giro van de federatie , met de vermelding van de naam van het artikel en uw lidmaatschapsnummer.

I N H O U D S O P G A V E

CONTRIBUTIE 1995

Nog voor een keer !? , Ja, alweer .!

Het bedrag van fl. 25,- over te maken naar :
gironum 5244293 t.n.v. Fed. v.Atomclubs Ned.en Belgie
Berkenlaan 24
3737 RN Groenekan.
Ofwel , als U een bankrekening wilt gebruiken, naar:
nummer P 5244293 t.n.v. Fed. v.Atomclubs , etc zie hierboven

***** Er bleken enkele leden te zijn die "snel lezen",; in het redactionele commentaar van Roland in het vorige vertelde hij " Ik zie het in gedachte al, de begeleidende brief , enz, enz ,"

Enkele leden meenden dat de gedachte van Roland al werkelijkheid geworden was. Gelukkig is dat niet zo en vragen wij U wederom om zelf het geld over te maken. Voor de goede orde : onze voorraad acceptgiro's is al enkele jaren uitgeput en de kosten voor aanmaak zijn véél te hoog voor het kleine aantal leden. *****

Het Bestuur van de Federatie

VAN DE REDACTIE

Tot ons aller verbazing, alweer een jaar om, en nog wel met een welgevuld Atom-Nieuws !.
Hulde aan allen die daaraan hebben bijgedragen , namen noemen is in dit verband onnodig, men bladere in de inhoudsopgave van de 4 nummers.

Het lijkt er bijna op , dat de hartekreet van Roland de vorige keer zowaar effect heeft gehad, ben benieuwd of de variatie aan onderwerpen en schijvers zo groot blijft als in dit nummer het geval is.

In ieder geval is het duidelijk dat de ATOM-in-PC een gouden greep geweest is, als geconstateerd kan worden dat 90 % van de leden een dergelijk apparaat heeft aangeschaft.
Van de tweede serie zijn op dit moment nog maar een paar exemplaren over, zo weinig, dat het niet de moeite waard is , hem op te nemen in de lijst van atom-artikelen.

Namens de redactie nogmaals : aan allen dank voor de bijdragen, plezierige feestdagen , een goed en gezond nieuw jaar en op naar een volgend , naar we hopen , interessant ATOM-JAAR 1995.

Bruno Tossaint.

REGIO-MEDEDELINGEN.

1. REGIO BRABANT-OOST

   Bijeenkomsten op het bekende adres :
   Adolf van Cortenbachstraat 92, Eindhoven, tel. 040-123231.
   Aanvang 13.30.u

2. REGIO LIMBURG-BELGIE

   Clubavonden in "Oos Kaar",Geldersestraat 43, tel 046-321378.
   op de 1e vrijdag van de maand.
   N.B. 20 januari , jaarvergadering ,met "nonnevotten".

3. REGIO DEN HAAG

   Alles op het nieuwe adres : Theo Waayer
   Hendrik v. Boeijenlaan 66, 2273 DC Voorburg,t. 070-3862504

4. REGIO ARNHEM e.o.

   Geplande bijeenkomsten Acorn Atom Club Regio Arnhem:
   ten huize van Henri Derksen, Bolwerk 25, 6811 JE ARNHEM
   op de derde woensdag van de maand.
   Telefoon: 085-455485, UniCorn BBS: 085-425506 xxxx/xxxx BPS 8N1

Ontwikkelingen Atom-in-PC

door roland leurs

Dit keer geen nieuwe spectaculaire ontwikkelingen, behalve dat Atom Decados al draait met de originele Atom Dos Controller-kaart en een oude bekende tekstverwerker is weer beschikbaar.

Atom I/O kaart

De I/O-kaart moet nog van een klein detail worden voorzien; het enable-signaal moet een open collector uitgang worden, zodat andere kaarten ook het signaal kunnen bedienen. Naar verwachting komt deze kaart in januari 1995 beschikbaar.

Atom disk-drive

De Atom FDC met drives is inmiddels ook aangesloten op de Atom-in-PC kaart. Maar er is een ontwerp- cq denkfout gemaakt bij het aansluiten van de videolatches. Voor de video-emulatie wordt gebruik gemaakt van de NMI. Op het moment van ontwerpen ging ik er van uit dat de NMI toch nooit gebruikt zou worden. Want wie wil nu 100 kb diskettes gebruiken in een tijd dat halve gigabyte harddisks al betaalbaar zijn. En zo is de "bediening" van de VideoNMI geen open collector maar gewoon een push/pull uitgang.
De 8271 controller krijgt het helaas door deze opbouw niet voor elkaar om de NMI ingang van de processor laag te trekken; met andere woorden er kan niet van/naar Atom disk gelezen/geschreven worden. En twee weken nadat ik de I/O kaart heb aangekondigd met de zin 'Nooit meer krassen en onderbreken en piggy-packen' komt Henk Bastings met een Shottky diode en stanleymes om de NMI alsnog van een soort open collector te voorzien. De drive werkt, maar dat krassen is niet de meest ideale oplossing.
We zijn nu bezig om te onderzoeken of het kan om één uitgang van een GAL als open collector of Tri-state uitgang te programmeren; en zo ja, hoe doe je dat? U wordt op de hoogte gehouden.

ED64 tekstverwerker

Onze oude bekende tekstverwerker is inmiddels ook beschikbaar voor Atoms in PC. Deze versie is geheel compatible met alle voorgaande Atom exemplaren. Maar om het ding toch wat nieuws mee te geven is de naam veranderd in:

Atom Word Master v2.00

Dit is van origine een aanpassing van EDIT80 voor de koppeling Atom-Electron. Aangezien een belangrijk gedeelte van de operating systeem software voor de Atom-in-PC daarvan afgeleid is, is de aanpassing van die tekstverwerker een fluitje van een cent. Bij deze dus beschikbaar.

Het programma is zowel in geassembleerde als source-vorm te vinden op Henri's BBS en natuurlijk op de regioschijf.

Terminalprogramma versie 3.xx

De ontwikkelingen van het terminalprogramma V3.xx zijn inmiddels in volle gang. Er zijn een aantal grafische mogelijkheden toegevoegd zoals het tekenen van cirkels, rechthoeken, driehoeken en lijnen. Verdere uitbreidingen zullen worden de mogelijkheden om met sprites te werken in extended video modi en (patroon) kleurroutines voor gesloten vlakken (vergelijk met het PAINT-statement uit het GAGS-rom.

Mocht u zelf nog leuke of nuttige ideeën hebben dan bel, schrijf of E-mail gerust.

Rest mij nog om u namens Pascalle en mezelf prettige feestdagen te wensen.

Met vriendelijke groeten,

Roland Leurs telefoon: 046-370650
Prins Mauritslaan 43 fidonet: 2:285/226.9
6191 EC Beek

Atomdata in de pers

Is de Atom-revival zodanig, dat de pers er aandacht aan besteed? Nee, nog niet. Dit gaat over het samenpersen of comprimeren van data.

Inleiding

Wie wel eens zijn licht opgestoken heeft in de MS-DOS wereld, kent misschien de compressie programma's PKZIP, ARC en ARJ. Dat zijn van die handige programma's die gebruikt kunnen worden voor het verkleinen van data.
In dit verhaal probeer ik u uit te leggen hoe het een en ander in elkaar zit. De programeerlustigen onder u, wordt een complete analyse aangereikt om een AtomZip te maken. Leuk om uw naam dan weer eens in het clubblad te zien.

Soorten compressie

Het comprimeren van gegevens geschiedt in software. Er is, in ieder geval voor de Atom, nog geen chipset aanwezig die dit doet. En zo als voor alles in dit leven geldt ook hier: alles heeft zijn prijs. Een snelle compressie heeft een mager resultaat; een fantastische compressie kost een heleboel tijd.
Er zijn twee soorten compressie: een lossy en een lossless. Wat in gewoon Nederlands betekent: eentje waar je wel informatie mee verliest en eentje waar je geen informatie mee verliest.
In eerste instantie zal een ieder geneigd zijn te denken: wat heeft een lossy, eentje waar je dus wat informatie mee verliest, nou voor zin. Wel wanneer het gaat om het overslaan van niet relevante informatie dan zal het duidelijk zijn dat datgene wat je niet nodig hebt, ook niet als ballast meegenomen hoeft te worden. Denk aan een foto. Als u een prachtige kleurenfoto scant en er een zwart-wit afbeelding van gaat maken, dan heeft het geen zin om kleureninformatie bij dat betreffende bestand op te slaan. Zo ook de lossy compressie.
Bij een lossless compressie gaat niets verloren, de uitkomst na de-compressie is identiek aan de invoer aan het compressie algoritme.

Datacompressie is niet zo'n wonder als het op het eerste ge-

zicht lijkt. Maar ook hier krijg je niets voor niets; de niet bestaande 'wet van behoud van informatie' wordt meestal niet overtreden. Denk aan schijfcompressie. Een (MS_DOS) programma als DoubleSpace kan meer bytes op schijf kwijt, omdat DOS slordig met de ruimte omspringt en te veel ruimte reserveert voor een bestand. Of bekijk onze Nederlandse taal: bepaalde letters en combinaties van letters komen vaker voor dan andere; met een beetje bijdehand coderen heb je minder bytes nodig dan wanneer je voor elk letterteken domweg één byte reserveert.

Drie algoritmen

Er bestaan een aantal zeer bruikbare algoritmen. Ieder met zijn specifieke voor- en nadelen. De drie meest populaire worden hier wat toegelicht

RLE

De eenvoudigste van het stel werkt op basis van Run Length Encoding. RLE is een loss-less methode; er gaat geen bit verloren. Stel je hebt een bestand wat een Clear 4 plaatje codeert. Ieder bit representeert een schermpuntje. Het bestand begint met een aantal regels waar geen puntjes staan, dus de eerste bytes bevatten allen 00. Stel dat dat 40 bytes zijn, dus 320 punten, dan kan je in plaats van 40 maal een 00 ook een codering (40) 00 aangeven. Deze opslag kost dan 2 bytes in plaats van 40. Het uitpak algoritme leest eerst het aantal (40) en daarna de waarde 00, die dus 40 keer moet worden geschreven.
Hoe meer naast elkaar gelegen bytes de zelfde waarde hebben, des te beter het RLE-algoritme presteert. Dat betekent dat de methode geschikt is voor plaatjes met weinig variatie. Voor rommelige, chaotische beelden heb je er niet veel aan.

LZW

Zo'n zelfde bezwaar geldt ook, hetzij wat minder, voor het LZW algoritme, een creatie van de heren Lempel, Zif en Welch. LZW wordt veel gebruikt. In TIFF en GIF plaatjes is de data met LZW gecomprimeerd, en ook het bekende inpakprogramma PKZIP past het toe.
Bij LZW worden tijdens het lezen van het invoerbestand een tabel opgebouwd met combinaties van invoerstrings die tot dan

toe zijn gelezen. Zodra een zelfde string (serie bytes dus) opnieuw optreedt, wordt alleen genoteerd op welke plek in de tabel die serie staat. Een verwijzing naar zo'n tabel is natuurlijk veel korter dan de serie bytes weer opnieuw noteren.
LZW is eveneens een loss-less algoritme en presteert beter naarmate het bestand meer regelmaat bezit.

Huffman

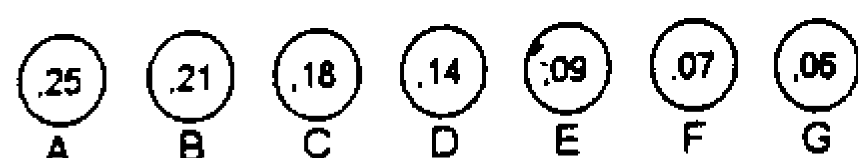
In 1951 was David Huffman student aan het Massachusetts Institue of Technology. De examenopgaven die hij van zijn leraar informatiekunde kreeg klonk eenvoudig: zoek de beste manier om gegevens in binaire code op te slaan. Hoe simpel ook omschreven, inhoudelijk was het een lastig probleem.
Zijn uitgangspunt was het feit dat het ene symbool vaker voorkomt dan het andere. Een foto van een zeegezicht bevat bijvoorbeeld meer blauwe tinten dan rode, en in een tekst struikel je vaker over de letter E dan de letter Q. De truc is dan om de E met minder bits te coderen dan de Q.
In een gewoon ASCII bestand wordt elke letter door één byte gecodeerd; het woord QEEQEE kost dus zes bytes. Geef je de E echter een kortere code dan de Q, bijvoorbeeld E=0.5 en Q=1.5, dan heb je nog maar vijf bytes nodig om het zelfde woord op te slaan. In een lange tekst, waar in verhouding meer E's dan Q's voorkomen, wordt de besparing nog groter.
Het idee was niet nieuw, maar Huffman ontdekte dat het handig was een boomstructuur van 'knopen' te bouwen, waaruit je altijd de aller gunstigste codering kon aflezen. Zijn ingeving was simpel: begin niet bovenaan te bouwen aan de boom maar onderaan. Echte bomen groeien ten slotte ook zo.

Zullen we de boom vast opzetten?

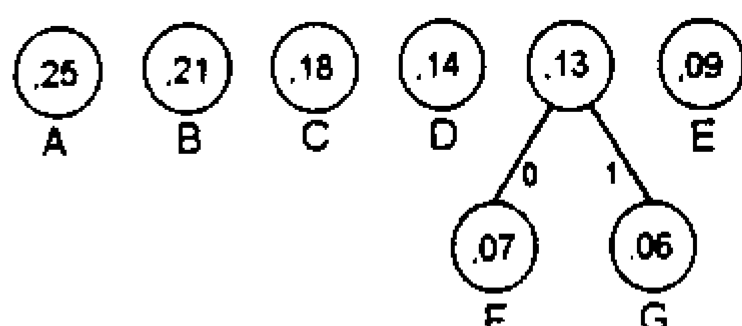
Laten we het bouwen van een Huffman Boom in een voorbeeld naspelen. De hier gekozen frequenties van letters komt niet overeen met de werkelijkheid. Wie echt zijn tanden in dit onderwerp wil zetten, wordt naar het boek 'The Turning Omnibus' van A.K. Dewdney verwezen. Maar let op: dat is veteranenkost. Voor het gemak worden de letters A,B,C,D,E,F en G gebruikt, die met de volgende frequenties in een tekst optreden: 0.25, 0.21, 0.18, 0.14, 0.09, 0.07 en 0.06. Bij elkaar opgeteld geven ze precies 1 ofwel 100%.
Om tekst volgens de Huffman-codering samen te persen, moet je dus eerste alle letters tellen zodat je de percentages kunt

vaststellen, ofwel de kans dat ze optreden. Als je dit te lang duurt, je moet tenslotte toch het hele bestand door lezen en tabellen maken, kun je de zaak bespoedigen door een standaard letterfrequentie te gebruiken. Voor iedere taal is zo'n frequentie tabel bekend.

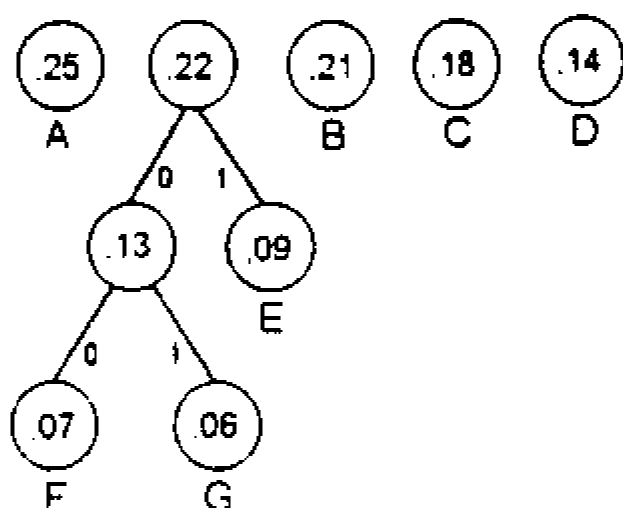
Het algoritme om de Huffman-boom te bouwen, werkt als volgt (zie ook de afbeelding): rangschik de letters naar oplopende frequentie (fase START) en voeg een nieuwe knoop toe (Stap 1), die 'ouder' wordt van de twee letters met de laagste frequentie.



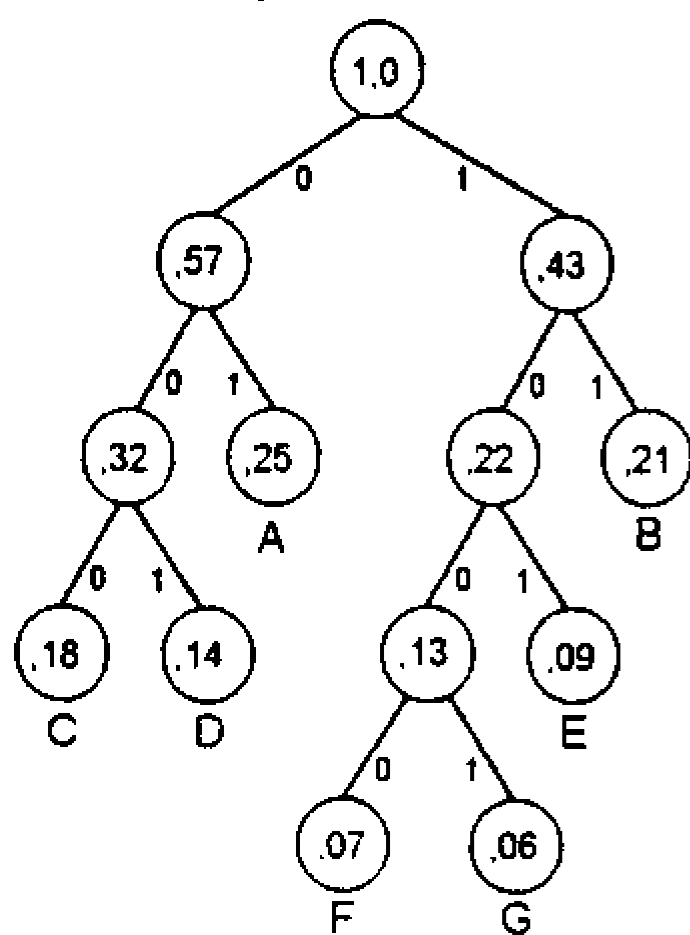
Start



Stap 1



Stap 2



Huffman-code

A 01
B 11
C 000
D 001
E 101
F 1000
G 1001

De aftakking naar links krijgt een 0 als code, die naar rechts een 1. De knopen worden vervolgens opnieuw gerangschikt (stap 2), waardoor de ouder van F en G op de een na laatste plaats komen. Herhaal dit proces tot dat alle letters een eigen eindknoop hebben. Na deze stappen heb je de complete Huffman-boom, waar bij alle letters een eindknoop zijn en de bovenste knoop een frequentie heeft van precies 1.0. De weg door de boom van boven naar beneden geeft nu aan welke codering je voor welke letter moet gebruiken. A wordt dus gecodeerd als 01, B als 11, tot aan G=-1001. De letters die het vaakst optreden, hebben de kleinst mogelijk codering gekregen.

Een blik in de recente toekomst: Fractale compressie

Fractale compressie wordt al commercieel toegepast. Het produkt Encarta van Microsoft, wat op CD-Rom beschikbaar is, gebruikt dit. Dit algoritme is een lossy algoritme maar heeft als belangrijk voordeel dat er geen lelijke blokken ontstaan, zoals dat bijvoorbeeld met de hier niet besproken JPEG en MPEG, maar dat het beeld in zijn geheel hooguit wat wazig wordt, alsof er met een grove penseel over is gestreken. Toch stoort dat juist minder.

Conclusie

Er zit nog veel muziek in compressie. Velen maken van dit soort zaken hun afstudeer opdracht. Met name met de komst van een traag medium als CD-Rom in combinatie met heel veel informatie, is het nodig dat er hoogstaande compressie algoritmen komen. Deze zijn soms al als chip set beschikbaar, maar feitelijk staat dat allemaal nog in de kinderschoenen. Toch zal het binnen 10 jaar mogelijk zijn om een 1,5 uur durende speelfilm op volle grootte en voorzien HiFi geluid op een 'normale' CD-Rom te krijgen.

INTIKKEN EN RUNNEN MAAR:

```
 10 PROGRAM KLOKrtc
 12 REM BERRY LAM 850428
 14 REM LEM DULSTRAAT 57
 16 REM 2801 EP GOUDA
 18 REM 01820-84619
 20 DIM G4,T8,KK3
 30 BASE #78
 40 DEF STIP,0011110000000000
 41 DEF:     0111111000000000
 42 DEF:     1111111100000000
 43 DEF:     1111111100000000
 44 DEF:     1111111100000000
 45 DEF:     1111111100000000
 46 DEF:     0111111000000000
 47 DEF:     0011110000000000
 50 ASSIGN STIP,0
 59 REM invoer begin-tijd
 60 TIME $G;G?2=G?3;G?3=G?4;G?4=13
 70 IF LEN G<>4;GOTO 60
 80 D=G?3-48
 90 IF D<0 OR D>9;GOTO 60
100 C=G?2-48
110 IF C<0 OR C>5;GOTO 60
120 B=G?1-48
130 A=?G-48
140 IF A<0 OR A>2;GOTO 60
150 IF B<0 OR B>9;GOTO 60
160 IF B>3;IF A=2;GOTO 60
169 REM teken begin-tijd
170 CLEAR 4
190 I=A;S=0;GOSUB 800
200 I=B;S=50;GOSUB 800
210 I=C;S=150;GOSUB 800
220 I=D;S=200;GOSUB 800
230 SET 0,114,30;SET 0,124,30
240 SET 0,114,50;SET 0,124,50
248 REM start tijd-lus
249 TIME $T; E=T?4; F=E
250 DO
255 DO TIME $T;E=T?4;UNTIL E<>F;F=E;REM wacht tot 1 minuut
verstreken
260 I=A;J=B;K=C;L=D
269 REM verhoog tijd 1 minuut en wijzig evt. ook andere
cijfers
270 D=D+1
280 XIF D=10;D=0;C=C+1
290  XIF C=6;C=0;K=10;B=B+1
300   XIF B=4 AND A=2;B=0;A=0;J=11
310   N=12;S=0;GOSUB 900
```

```
  320    ELSE REM
  330    XIF B=10;B=0;A=A+1
  340    N=I;S=0;GOSUB 900
  350    ELSE REM
  360    N=J;S=50;GOSUB 900
  370   ELSE REM
  380   N=K;S=150;GOSUB 900
  390 ELSE REM
  400 N=L;S=200;GOSUB 900
  410 UNTIL 0
  799 REM teken-routine begin-cijfers
  800 RESTORE(1000+I*10)
  810 DO READ Y
  820 IF Y=0;GOTO 850
  830 READ X;X=X+S
  840 SET 0,X,Y
  850 UNTIL Y=0
  860 RETURN
  899 REM routine om de cijfers te laten verspringen
  900 RESTORE(2000+N*10)
  910 DO READ Y
  920 IF Y=0;GOTO 990
  930 READ X;READ Z;READ W
  940 X=X+S;Z=Z+S
  950 V=SGN(Z-X)
  960 IF W=1;SET 0,X,Y;SET 0,X,Y;DO CARRY 0,X,Y;X=X+V;UNTIL
X=Z+V;SET 0,Z,Y
  970 IF W=2;SET 0,X,Y;SET 0,X,Y;DO CARRY 0,X,Y;X=X+V;UNTIL
X=Z+V
  980 IF W=3;SET 0,(X+V),Y;DO CARRY 0,X,Y;X=X+V;UNTIL X=Z+V
  990 UNTIL Y=0
  995 RETURN
  999 REM data voor het tekenen van de begin-cijfers
 1000 DATA 10,14,10,24,20,4,20,34,30,4,30,34,40,4,40,34
 1001 DATA 50,4,50,34,60,4,60,34,70,14,70,24,0,0
 1010 DATA 10,24,20,24,30,24,40,24,50,24,60,24,70,24,0,0
 1020 DATA 10,4,10,14,10,24,10,34,20,4,30,4,40,14,40,24
 1021 DATA 50,34,60,4,60,34,70,14,70,24,0,0
 1030 DATA 10,14,10,24,20,4,20,34,30,34,40,14,40,24
 1031 DATA 50,34,60,4,60,34,70,14,70,24,0,0
 1040 DATA 10,34,20,34,30,34,40,4,40,14,40,24,40,34
 1041 DATA 50,4,50,34,60,4,60,34,70,4,70,34,0,0
 1050 DATA 10,14,10,24,20,4,20,34,30,34,40,14,40,24
 1051 DATA 50,4,60,4,70,4,70,14,70,24,70,34,0,0
 1060 DATA 10,14,10,24,20,4,20,34,30,4,30,34,40,4,40,14,40,24
 1061 DATA 50,4,60,4,60,34,70,14,70,24,0,0
 1070 DATA 10,4,20,4,30,14,40,24
 1071 DATA 50,34,60,34,70,4,70,14,70,24,70,34,0,0
 1080 DATA 10,14,10,24,20,4,20,34,30,4,30,34,40,14,40,24
 1081 DATA 50,4,50,34,60,4,60,34,70,14,70,24,0,0
 1090 DATA 10,14,10,24,20,4,20,34,30,34,40,14,40,24,40,34
 1091 DATA 50,4,50,34,60,4,60,34,70,14,70,24,0,0
 1999 REM data voor het verspringen van de cijfers
```

```
2000 DATA 10,14,24,1,20,34,24,2,20,4,24,1,30,34,24,2
2001 DATA 30,4,24,1,40,34,24,2,40,4,24,1,50,34,24,2
2002 DATA 50,4,24,1,60,34,24,2,60,4,24,1,70,14,24,1
2003 DATA 0,0,0,0
2010 DATA 10,24,34,3,10,24,14,3,10,14,4,3,20,24,4,2
2011 DATA 30,24,4,2,40,24,14,3,50,24,34,2,60,24,4,3
2012 DATA 60,24,34,2,70,24,14,3,0,0,0,0
2020 DATA 10,34,24,1,10,4,14,1,20,4,34,3,30,4,34,2
2021 DATA 0,0,0,0
2030 DATA 10,14,24,1,10,24,34,2,20,4,34,1,40,14,4,3
2031 DATA 40,24,34,3,50,34,4,3,70,14,4,2,70,24,34,2
2032 DATA 0,0,0,0
2040 DATA 10,34,24,2,10,24,14,3,20,34,4,3,40,4,14,1
2041 DATA 40,34,24,1,50,34,4,1,60,34,4,1,70,4,14,3
2042 DATA 70,34,24,3,0,0,0,0
2050 DATA 30,34,4,3,40,14,4,3,60,4,34,3,70,4,14,1
2051 DATA 70,34,24,1,0,0,0,0
2060 DATA 10,24,14,1,10,14,4,2,20,34,4,1,30,4,14,2
2061 DATA 30,34,14,1,40,4,14,1,40,14,24,1,50,4,34,2
2062 DATA 60,4,34,1,70,14,4,3,70,24,34,3,0,0,0,0
2070 DATA 10,4,14,2,10,14,24,3,20,4,34,3,30,14,4,3
2071 DATA 30,14,34,2,40,24,14,3,50,34,4,3,60,34,4,3
2072 DATA 70,4,14,1,70,34,24,1,0,0,0,0
2080 DATA 30,4,34,1,40,24,34,3,0,0,0,0
2090 DATA 30,34,4,3,40,24,14,1,40,14,4,2,0,0,0,0
2100 DATA 30,34,4,3,40,14,4,2,40,24,34,2,50,4,34,3
2101 DATA 60,4,34,3,70,4,14,1,70,34,24,1,0,0,0,0
2110 DATA 30,34,4,3,40,14,4,2,40,24,34,2,50,34,4,3
2111 DATA 0,0,0,0
2120 DATA 10,4,14,1,10,34,24,1,20,4,34,3,30,4,34,3
2121 DATA 40,14,4,2,40,24,34,2,50,34,4,3,0,0,0,0
```

Een verhaal over het maken van een verhaal

Inleiding

Nee, nee, geen uitgebreid verhaal over het recursief programmeren zoals de titel wellicht zou vermoeden, maar een verhaal over hoe je een stuk informatie zodanig op papier krijgt, dat je medemens daar ook nog iets zinnigs van kan opsteken.
Naar aanleiding van de (terechte) dramatische oproep van Roland in het vorige nummer, even een handreiking aan hen, die wel willen schrijven maar even niet weten waar te beginnen.

De opzet

Kent u dat gevoel, wat u krijgt als zo'n leeg vel of scherm u staat aan te staren. Waar moet je beginnen en waar moet je eindigen. Ik heb mij door een gerenommeerd aannemer eens laten vertellen dat hij dat gevoel ook heeft. Daar sta je dan in de polder, tot je enkels in de bagger. Niets maar dan ook niets duidt er op dat daar binnen 18 maanden een High-tech gebouw zal komen te staan. Hij werkt echter met gereedschappen zoals tekeningen en planningen. Zonder dit gereedschap, is het project gedoemd te mislukken. Zo is het ook met het schrijven van een stukje: zonder een juiste aanpak is het nauwelijks tot een goed einde te brengen.

Volgorde

Volgorde is belangrijk. Zoals ook een aannemer niet de muren gaat metselen voordat er een fundering ligt, zo heeft ook een verhaal een bepaalde opbouw: een inleiding, de clou en een einde, niet noodzakelijkerwijs in die volgorde, maar dat is wel praktisch. Vertel in de inleiding waar het over gaat: is het hardware beschrijf dan middels een pakkend voorbeeld waar het over gaat. Mooi voorbeeld bij het aansturen van een display is het schermpje met de 'ROM=PCHARM' 'WriteProtect = ON' uit het verhaal van Guido in de vorige AN.
Gaat het over software, vertel dan wat het doet en wat de reden is om er over te schrijven. Dat kan twee doelen hebben, namelijk het uitleggen van de gebruikte techniek of het begeleiden van de gebruiker van het programma als een extra vorm van handleiding.
Het verhaal wat u schrijft, heeft een doel. Zorg dat in de

inleiding al duidelijk wordt, wat de bedoeling is. Lukt dat daar niet, dan is het gedoemd te stranden op het niveau van bladvulling.
Wie de lezer bij de inleiding boeit, heeft hem vast tot aan de laatste regel van het stuk, ook al zijn er passages die niet na één keer lezen duidelijk zijn.

De clou

Waar het echt op aankomt, wordt in de middenmoot van het stuk beschreven. Let er op dat dit onderverdeeld wordt in hanteerbare brokken. Zelf gebruik ik zo maximaal 18 regels per item. Wordt dit langer, dan probeer ik het op te knippen in nog kleinere items.
In de clou wordt iets over de details uitgelegd. Waarom moet pootje 14 aan de plus hangen of waarom is het handig om die instructies in die specifieke volgorde te gebruiken.
Gebruik gewone Nederlandse taal. Het heeft geen zin om een verhaal in de stijl van Karel ende Elegast te schrijven. Zo praten we tenslotte ook niet met elkaar.
Gebruikt u een programmalisting, gebruik dan gerust een stuk daarvan in de tekst om dit nader uit te leggen. Zorg daarbij wel voor voldoende leesbaarheid. Iedereen weet wel dat PRINT tot P. kan worden afgekort, maar PRINT is nou eenmaal toch wat duidelijker. En in een tijd waar geheugen tot in voldoende mate beschikbaar is, is dat niet echt een argument meer.

De staart

Het ei is gelegd. Eindelijk een inleiding met clou van één of meer pagina's. En dan nog even de afronding. Die kan kort zijn. Prikkel de lezer tot nader experimenteren. Tracht de medemens uit de tent te lokken om er zelf ook iets aan te doen. Het is best leuk om te zien dat anderen dat oppakken.
Het doet mij bijvoorbeeld veel plezier als ik mijn roemruchte assemblerversie zie van de CRC berekening van de ROMS. Zelfs in het afgelopen nummer kwam ik deze routine 11 (!) jaar na dato weer eens in een uitstekend programma tegen. Zoiets is nou exact wat ik bedoel. Dat wat je schrijft heeft een doel en kan door anderen worden geïmplementeerd in een groter geheel.

Waarover kan je schijven

Het schrijven van een verhaal is in basis niet echt moeilijk. Als het de moeite waard is om er een of meerdere avonden mee bezig te zijn, dan is het ook de moeite om er een verhaaltje over te schrijven. Niet alle verhalen hoeven over uitgesproken high tech dingen te gaan. Hoe bijvoorbeeld een spel als Invaders werkt, dat is best leuk. Of wat je met de Atom doet. Hoe heb je de software verdeeld over tapes of diskettes.
Gebruik als 't effe kan, wat humor in het verhaal. Die nulletjes en eentjes zijn al droog genoeg, dus waarom niet eens een gekke opmerking.

De layout

Het verhoogt de leesbaarheid als gebruik wordt gemaakt van kleine hoofdstukjes met een titel. Niet te lang en niet te ingewikkeld.
Probeer de conditie van uw printer aan te passen aan de leesbaarheid. Beter een verhaal op disk opgestuurd naar de eindredacteur, dan een donkergrijze letter op een stuk kringlooppapier.
Listings mogen wat mij betreft nooit meer dan 2 pagina's in beslag nemen. Daarna haakt de lezer toch af. Beter is het dan om de programma's via de bestaande distributie kanalen te verspreiden en de elementaire zaken in het stuk te noemen.
Gebruikt u plaatjes of tekeningen, probeer dan duidelijk over te komen.

Wees niet bang om fouten te maken

Het gaat ook wel eens mis. Heeft u het verhaal gelezen in het vorige nummer, over het trekken van lijnen? Nou, daar zaten plaatjes bij, gemaakt in Corel Draw. Heeft 2 avonden gekost. Uitgeprint op een sjonge-wat-is-ie-duur postscriptprinter van het bedrijf. Het zag er qua lay-out goed uit. Maar het uiteindelijke resultaat in Atom Nieuws: ik kon zelf amper lezen wat er staat. Oorzaak: de lijntjes waren te dun, en de afbeelding op A4 terwijl het boekje A5 is.
Is het verhaal eenmaal uit de printer gerold, laat het dan door iemand lezen die u verdenkt van enig taaltechnisch inzicht. Het verwijderen van tik- en stijlfouten is voor de niet-schrijver van dat stukje veel eenvoudiger dan voor de

schrijver zelf. Echt, u behoort tot de uitzonderingen als u uw eigen fouten kunt vinden.

Wees niet bang

Het maken van fouten is niet erg, het maken van twee keer dezelfde fout wel. Daarom is het van belang dat u iets schrijft. Zo kan de kennis binnen de club worden vergroot. Maar er is meer. Zo bouwt u ook aktief mee aan het ontwikkelen van uw eigen persoonlijkheid. Communicatie tussen geest en papier, of in gesproken woord, is een van de belangrijkste dingen in het maatschappelijk functioneren. Lijkt wel een zin uit een commerciele folder voor een training. Dat kon ook wel eens zo wezen maar is, ondanks dat, toch meer dan waar.
Het feit dat u bijvoorbeeld de hier geschetste indeling overneemt is niet een kwestie van plagiaat of iets dergelijks. Ik heb zelf de 6502 niet bedacht, toch kan ik er leuke dingen mee doen. Dat is dus precies het zelfde. Gebruik het raamwerk Inleiding, Clou en conclusie om het verhaal op te zetten. Bedenk wat u met het verhaal wilt bereiken. Ga er voor zitten en u zult zien: de eerste keer een drama, de tweede keer is het al eenvoudiger en vanaf de derde keer is het leuk. Vraag dat maar eens aan de overige schrijvers in de Atom Nieuws.
Hoe dat er in de praktijk uit kan zien, bekijk dat maar eens elders in het nummer, naar het stukje over 'de schoonheid van tabellen'.

Conclusie

Het is goed te doen om stukjes te schijven. Het zal altijd zo zijn dat er verschil is tussen de ene en de andere schrijver, maar voor mensen binnen deze club moet het toch allemaal mogelijk zijn om iets op papier te krijgen. En dat hoeft ook niet eens echt veel tijd te kosten. Als je van te voren weet, welke indeling je aanhoud, is het in een uur te realiseren. Dit verhaal bijvoorbeeld is in 3 kwartier bedacht en gerealiseerd. Reken daar 10 minuten bij voor het nakijken en corrigeren van stijlfouten en typo's. Moet te doen zijn dus.
Nog wat ideeën om over te schijven nodig: Wie legt uit hoe Space Invaders is opgezet. Op het scherm gebeuren een heleboel dingen te gelijkertijd. Hoe doe je dat?
Wie vertelt iets over de Floating point rom?
Van de Atom je werk gemaakt? Of mede dankzij de Atom de huidi-

ge baan veroverd. Vertel eens hoe dat gaat.
Iets beleefd met je Atom waar je niet uitkomt? Schijf eens helder op papier wat je gedaan hebt en vraag het antwoord aan je mede Atomisten.
Het blad bestaat bij de gratie van de schrijvers. Het zou toch zonde zijn als een blad als deze moet opdoeken door het gebrek aan kopij. De Atom is toch nog steeds zodanig interessant, dat het 25 jarig bestaan van de club moet gehaald kunnen worden. Wellicht tegen die tijd op een kaart met optische verbinding naar een machine met bio geheugen, draaiend op een kloksnelheid in de buurt van het optische spectrum. Beeldscherm en toetsenbord zijn dan alleen nodig voor de Atom-in-?? kaart. De andere interface geschiedt middels een RS-464 connector in het achterhoofd..... en vooral een multi-media uitgave van Atomnieuws. Ieder kwartaal opnieuw de mooiste stukjes van het die de stap op het schrijverspad hebben gezet, naar aanleiding van een artikel in de jaren '90.......

Leendert

De schoonheid van tabellen

Het nut van een tabel

Iedereen kent natuurlijk tabellen. Of het nu gaat om het berekenen van een sinus of om te bepalen hoeveel calorieën er in de vakantie per eenheid produkt zijn bijgekomen: tabellen zijn goed voor het betere opzoekwerk.
Tabellen kunnen ook voor het versnellen van programma's worden gebruikt. Op die manier wordt eenmalig rekenwerk opgeslagen in een tabel en middels het raadplegen van die tabel wordt snelheid bereikt.

Wat gaan we maken

Tabellen heeft altijd de associatie met saai. Vandaar dat ik mijn best gedaan heb een sprankelend voorbeeld te bedenken. Wat we gaan doen is het volgende: Beeldschermmode 4 wordt gespiegeld rondom zijn verticale midden-as. Met andere woorden: wat links staat, staat dan rechts en wat rechts staat, dat staat dan links, maar dan beide in spiegelbeeld. Enfin. Wie niet weet wat spiegelbeeld is, moet zich nog maar eens gaan scheren, met excuus aan de vrouwelijke lezer.

De beeld opbouw

Natuurlijk volkomen overbodig, maar omdat de Atom Nieuws anders toch niet volkomt, hier een kleine samenvatting van hoe Clear 4 werkt.
Iedere regel bestaat uit 32 bytes, die ieder goed zijn voor acht individuele puntjes. Een bitwaarde 1 komt overeen met een oplichtend puntje, de bitwaarde 0, schop open die deur, met de afwezigheid van een oplichtend puntje.
Afhankelijk van de mode, zijn er tot 192 beeldlijnen. Dus 192 maal 32 bytes.

Het Spiegelen

Het spiegelen van een scherm, wil zeggen dat wat links komt te staan en andersom. Maar het niet eenvoudig het verwisselen van de twee bytes. Om dit te illustreren hier een bitpresentatie

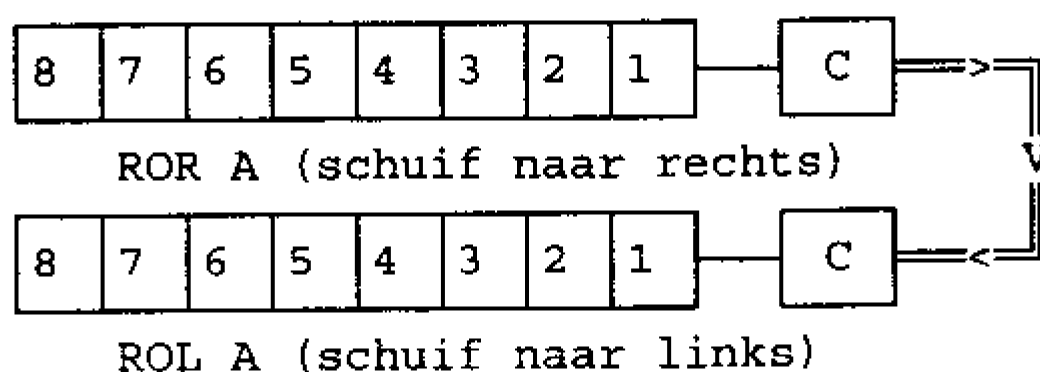
van een paar puntjes. 0010.0000 wordt in spiegelbeeld 0000.0100. Wat moet er verder gebeuren. Wel het wisselen van individuele punten is een tijdrovende aangelegenheid. Beter is het om het op byte niveau, dus in Clear 4 met acht punten tegelijk. Hoe ziet dat er in de praktijk uit. Wel eerst wordt de waarde van #801F, wat overeenkomt met de acht puntjes in de rechterbovenhoek van een Clear 4 scherm, opgehaald. Deze wordt gespiegeld en even opgeslagen. Vervolgens wordt #8000, de acht bytes uit de linkerbovenhoek opgehaald en na gespiegeld te zijn, geplaatst op locatie #801F. Het tijdelijk bewaarde resultaat van het voormalige #801F wordt weer opgehaald en geplaatst op locatie #8000. Vervolgens wordt verder gegaan met #801E en #8001. Dit gaat zo door tot #800F en #8010. Dat is dus de spiegellijn.

Het spiegel algoritme

Het verplaatsen van individuele bits gaat met een rotatie instructie. Om een spiegelbeeld van een willekeurig byte te krijgen kan met de volgende methode worden gewerkt.

```
   for n=1 to 8
     Carry is ror waarde
     Spiegel is rol waarde met Carry
   next
```

Een beetje abacradaba op het eerste gezicht. Maar wat gebeurt er. Het byte wordt acht maal naar rechts geschoven, waarbij het resultaat in de Carry komt te staan. Dit resultaat wordt vervolgens naar links geschoven met de Carry. Op die manier ontstaat een spiegelbeeld van de invoerwaarde. Nog even een plaatje:



Tot zover niets aan de hand. Het werkt naar behoren. Even wat tellertjes om het geheel van #8000 tot #97FF te laten geschieden en klaar is Leen. Maar. Maar dat gaat allemaal niet zo snel, want per spiegel zijn er een aantal instructies nodig:

```
          ldx @8

again     ror a
          rol #70
          dex
          bne again
```

Per byte altijd nog goed voor 8 * (2+5+2+3) clock cycli. Dit moet voor steeds paartjes van twee bytes, u weet wel #8000 en #801F gebeuren dus per wissel gaan er 192 clock cycli in CPU-rook op.
De oplettende lezer zal al vermoeden dat dit op een andere manier kan.

Oplossen met een tabel

Als gekeken wordt naar de mogelijkheden om een byte te spiegelen, dan zijn er waarden van 0 tot 255 met een bijbehorend spiegelbeeld. Wat is dan eenvoudiger om een tabel te maken, waar in op locatie X, waarbij X mag liggen tussen 0 en 255, de gespiegelde antipode staat opgeslagen. Dat scheelt veel overbodig rekenwerk.
Het maken van zo'n tabel is in assembler een koud kunstje. In Basic gaat dat al een stuk lastiger. Laat ons zien:

```
 10 REM SPIEGEL
 20 DIM K 255, LL20
 40 FOR N=1 TO 20; LLN=-1; NEXT N
 50 FOR N=0 TO 2 STEP 2
 60 P=#7000;[
 70 :LL0 LDX @#00
 80 :LL1 LDY @#08
 90 STX #72
100 :LL2 ROL #72
110 ROR A
120 DEY
130 BNE LL2
140 STA K,X
150 DEX
160 BNE LL1
170 RTS ; ]
180 NEXT
190 END
```

Er wordt ruimte gevraagd, middels de DIM k 255 voor een tabel met die omvang. Vervolgens wordt in een lus van het X register de hele zaak van 00 tot 255 opgebouwd. Locatie #72 wordt als hulpvariabele gebruikt en het Y register voor de acht bit posities die de waarden kunnen innemen.
Is zo'n tabel eenmaal een feit, dus als deze geïnitialiseerd is door het aanroepen van LL0, dan kan deze in het daadwerkelijke spiegelprogramma worden gebruikt.

De harde lus voor het spiegelen

Voor het wisselen van de waarden op de lokaties #8000 en #801F komt onder normale omstandigheden heel wat testwerk kijken. Niet bevorderlijk voor de snelheid dus. Vandaar dan gekozen is voor een andere opzet. Hiervoor wordt in hoge mate de stack misbruikt. Maar wat maakt het uit. Deze is ervoor en er is toch al voor betaald.
Wat er gebeurt is het volgende. De waarden van #801F tot en met #8000, worden in aflopende volgorde op de stack gezet.
Vervolgens wordt in de laatst weggeschreven waarde, die van #8000 dus, via de reeds genoemde tabel even gespiegeld en weer op locatie #801F weggeschreven. Niets klungelen met allerlei ingewikkelde testjes, nee, gewoon gebruik maken van het First in Last out principe van de stack. Even een weetje dus. Laat ons de definitieve code bekijken, inclusief de tabel en de initialisatie routine.

```
 10 REM SPIEGEL in 200 milliseconde
 20 DIM K 255, LL20
 40 FOR N=1 TO 20; LLN=-1; NEXT N
 50 FOR N=0 TO 2 STEP 2
 60 P=#7000;[
 70 :LL0 LDX @#00
 80 :LL1 LDY @#08
 90 STX #72
100 :LL2 ROL #72
110 ROR A
120 DEY
130 BNE LL2
140 STA K,X
150 DEX
160 BNE LL1
170 RTS
```

```
180 :LL6 LDA @#80
190 STA #71
200 LDY @#00
210 STY #70
220 LDX @24                   doe dit feest voor 6Kb dus 24
                              blokken van ieder 255 bytes
230 STX #72
240 :LL3 LDY @#1F
250 :LL4 LDA (#70),Y
260 PHA                       zet 32 bytes op de stack
270 DEY
280 BNE LL4
290 LDY @#1F                  Het byte rechts op het scherm
300 :LL5 PLA                  haal hem van de stack
310 TAX                       gebruik x als offset
320 LDA K,X                   Houdt hem tegen de spiegel
330 STA (#70),Y               en prak hem weer in het video
                              geheugen
340 DEY
350 BNE LL5
360 LDA #70
370 CLC
380 ADC@#20                   de volgende regel
390 STA #70
400 BCC LL3                   Geen page overgang
410 INC #71
420 DEC #72                   alle 24 blokken gehad?
430 BNE LL3
440 RTS;]
450 NEXT
460 LINK LL0 ;                REM initialiseer tabel
470 CLEAR 4
480 MOVE 10,10;DRAW 256,192; REM doe iets, kreng
490 DO
500   LINK LL6                REM doe iets in een lus
510 U.0
```

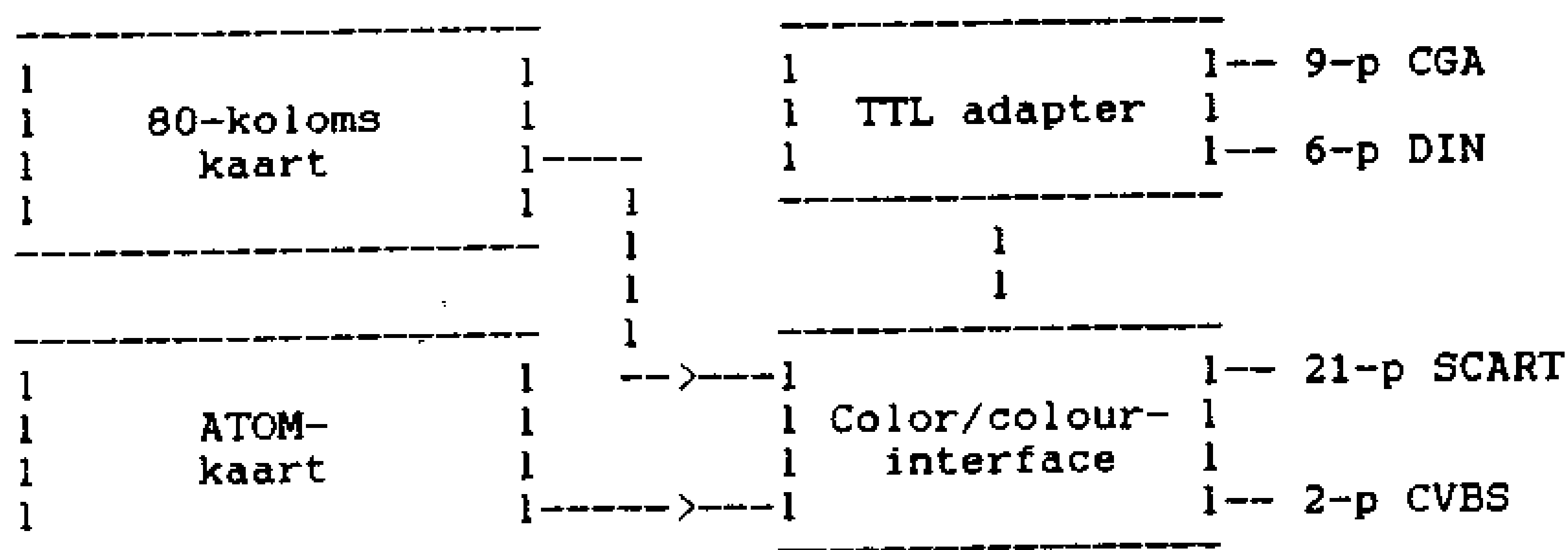
Conclusie

Zo ziet u hoe het mogelijk is om door middel van een tabel, toch hele snelle dingen te bereiken. Uiteraard onder vermelding dat het op een Intel op 100 MHz toch nog wat vlotter gaat. Maar daar hadden we het niet over.
Wie kan nog sneller spiegelen? Meldt het in de Atom, of leg het me uit, dan schijf ik er wel weer een stukje over!

===
= A T O M F U L L C O L O R / C O L O U R =
===

Deel 3

Ja, ja, ik weet het, deze kaart had mijnheer Acorn 10 jaar geleden moeten maken, maar helaas. Ze hebben daar wel een poging toe gedaan maar dat is zoals jullie weten niet zo goed gelukt. De problemen bij die kaart zaten in het digitalizeren van de analoge R-Y, B-Y en Y signalen, welke gemaakt worden door de videoprocessor MC6847. De pixelbreedte (tijd) was niet voor alle kleuren gelijk, en zeker niet als kleurenset 2 (met bit 3 van #B002) ingeschakeld werd. Bovendien was de kleurenuitcodering gedaan met meerdere poorten achter elkaar en ook nog per kleur een verschillend aantal in serie. Hierdoor ontstonden extra tijdverschillen tussen de kleurensignalen, waardoor er vieze gekleurde overgangen op het scherm te zien waren. Bovendien was er dan nog het bekende ruisprobleem. Kortom, de kaart werkte niet! Nog even ter herinnering het blokschema van het complete color/colour systeem. Het blokschema van het geheel zag eruit als volgt:



In de twee vorige delen (A.N. 10-2 en A.N. 12-3) zijn de 80-kolomskaart, de color/colour interface en de TTL adapter uitvoerig beschreven. In dit deel komt dan eindelijk de ATOM-kaart aan de beurt. De belangstelling zal waarschijnlijk wel nihil zijn, maar ik wil toch een poging wagen om uit te leggen hoe het een en ander werkt.

De ATOM kleurenkaart.

De opbouw van de kaart is als volgt:

a) Video processor U13 type MC6847.
b) A/D omzetters U10A t/m U12B type LM319.
c) RGBI codering U1 type 82S129.
d) RGBI latch U2 type 74LS379.
e) Sync scheider U10B, U6E en U6F type LM319 resp. 74LS04.
f) Pixelclock 3.58 resp. 7.16 MHz U7C en U7D type 74LS86.
g) Schakelsignaal decodering PC0, PC3, A/G en GM0 met U4A, U5A, U6A en U6D type 74LS74, 74LS02 en 74LS04.
h) Omschakelaar 6502 clock 1.0 naar 1.79 MHz U14A, U14B en U7B type 74120 en 74LS86.
i) Autosync (ruisonderdrukker) U4B, U8 en U9 type 74LS74, 74LS73 en 74LS02.
j) Signaalbuffers en kaartomschakeling U3 type 74LS367.

In het bijgaande schema is bovenstaande opsomming goed te herkennen en hierna zal ik punt voor punt trachten de werking enigzins te verklaren.

a) Video processor MC6847.

De video processor doet zijn normale werk zoals hij altijd al gedaan heeft in de ATOM, alleen voor deze schakeling verzorgt hij nog enige andere taken. Allereerst de drie video signalen Y, R-Y en B-Y en het referentie signaal CHB. Deze videosignalen zijn de signalen waar het in dit ontwerp om draait. In de ATOM worden deze signalen naar PL4 gevoerd, maar verder niet gebruikt. Verder maakt de chip nog een paar nuttige schakelsignalen, zoals A/G (Alpha/Graphics), GM0 (Graphics Mode), FS (Fieldsync) en DA0 (Data adres 0). Ook krijgt de videoprocessor nog signalen toegevoerd, vanaf de hoofdprint, welke in de kaart ook gebruikt worden. Dit zijn MS (Memory Select) en CLK47 (3.58 MHz clock).

b) A/D omzetters.

In de schakeling worden de videosignalen naar de A/D omzetters gevoerd. Deze omzetters zijn gewone comperators, dus eigenlijk 1 bit omzetters. Het analoge Y signaal op pin 28 wordt in U10A omgezet in een 1 bit digitaal Y signaal. Het beslissingsniveau wordt ingesteld met R19 voor monochroom. Als naar kleur wordt geschakeld vindt nog een correctie plaats met R21/T3. Dit is nodig omdat het DC-niveau van het Y signaal verandert bij omschakelen. Nu even goed opletten! Het referentiesignaal CHB wordt rechtstreeks aan U12A en U11A toegevoerd. Via spanningsdeler R14/R15 wordt een ongeveer 30% lagere referentie aan U12B en U11B toegevoerd. Het R-Y (PA) signaal wordt via spanningsdeler R12/R13 aan U12A en aan U12B en het B-Y (PB) signaal via R16/R17 aan U11A en U11B toegevoerd. De spanningsdelers zijn zodanig dat het R-Y resp. B-Y signaalniveau ongeveer qua niveau ligt tussen de waarde CHB op pinnen 5 en de 30% lagere waarde op pinnen 10 van U11 en U12. Dit betekent dat R-Y waarden die boven referentie CHB uitkomen een waarde 1 geven op de uitgang van U12A, dit is digitaal signaal A, en R-Y waarden die boven de CHB-30% uitkomen, een 1 geven op de uitgang van U12B, dit is digitaal signaal B. Voor het B-Y signaal geldt een zelfde redenering zodat op de uitgangen van U11A digitaal signaal C en van U11B digitaal signaal D staan. In figuur 10 uit het Motorola Data boek heb ik de referentieniveau's en de bijbehorende codering bijgetekend.

c) RGBI codering.

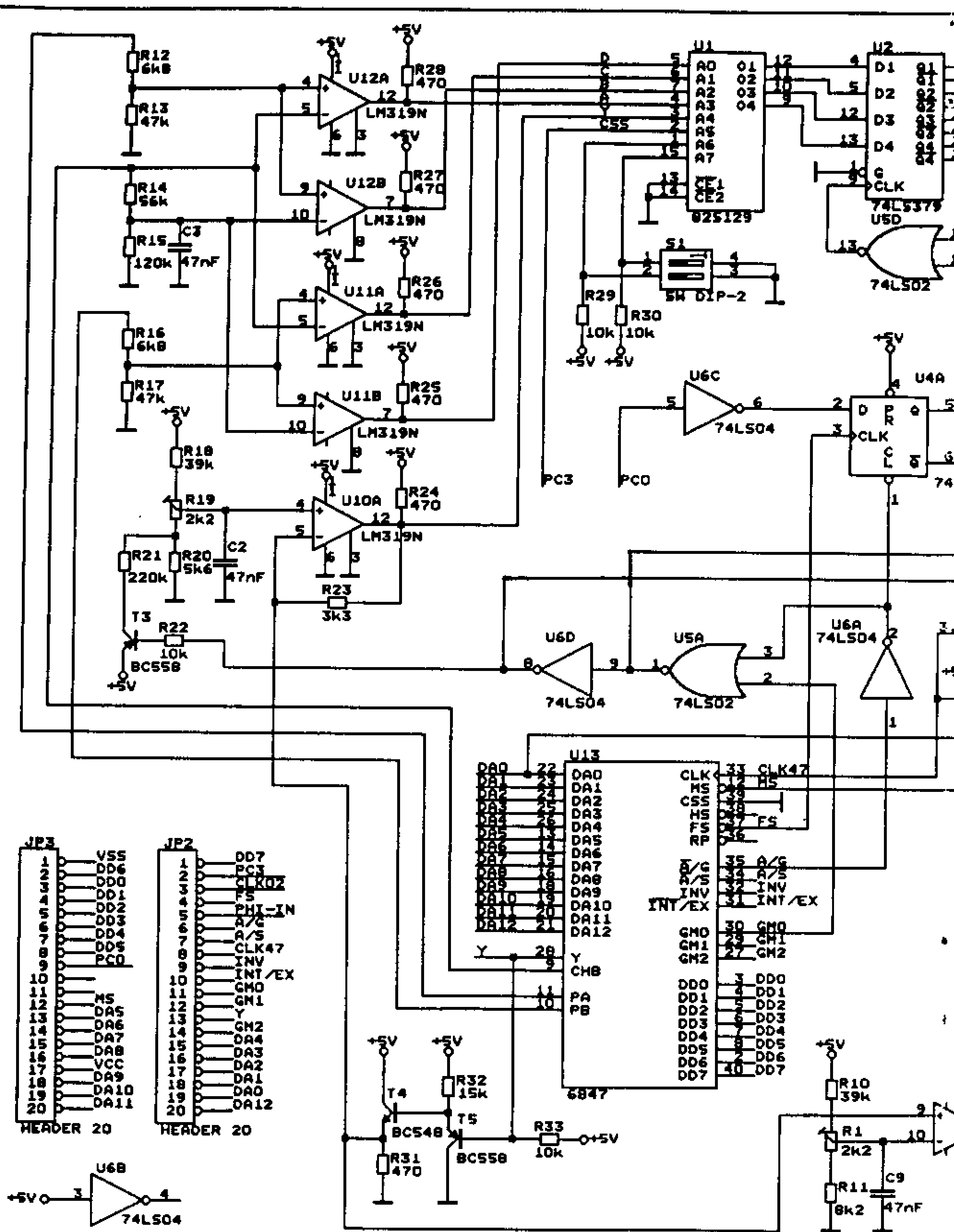
In de originele ATOM kaart werd de codering van de digitale signalen ABCDY met poortlogika gedaan met de bovenvermelde bezwaren. Ook werden de digitale signalen niet geklokt, zodat de pixelbreedte niet gegarandeerd was. Een en ander heeft mij tot de volgende oplossing gebracht: De gedigitalizeerde signalen worden in een PROM type 82S129 omgezet naar de RGBI signalen, de vertraging is dan voor alle signalen gelijk. Omdat de PROM (256*4) 8 adreslijnen en 4 outputs heeft, heb ik daar ook nog gebruik van kunnen maken door de colour set select CSS ook in de PROM te coderen. De CSS van de 6847 komt dan aan aarde te liggen zodat colour set 1 in de chip altijd is ingeschakeld. De omschakeling naar colour set 2 gebeurt dan dus in de PROM door colour set 1 naar set 2 om te coderen. Dit is veel beter daar de analoge omschakeling in de 6847 een DC shift gaf en daardoor het beslissingsniveau van de comparators beinvloedde. Dan zijn er nog 2 adreslijnen over waar de kleurensets mee gekozen kunnen worden. Zie tabel 1 waar de gehele codering goed is te overzien.

De codetabel 1.

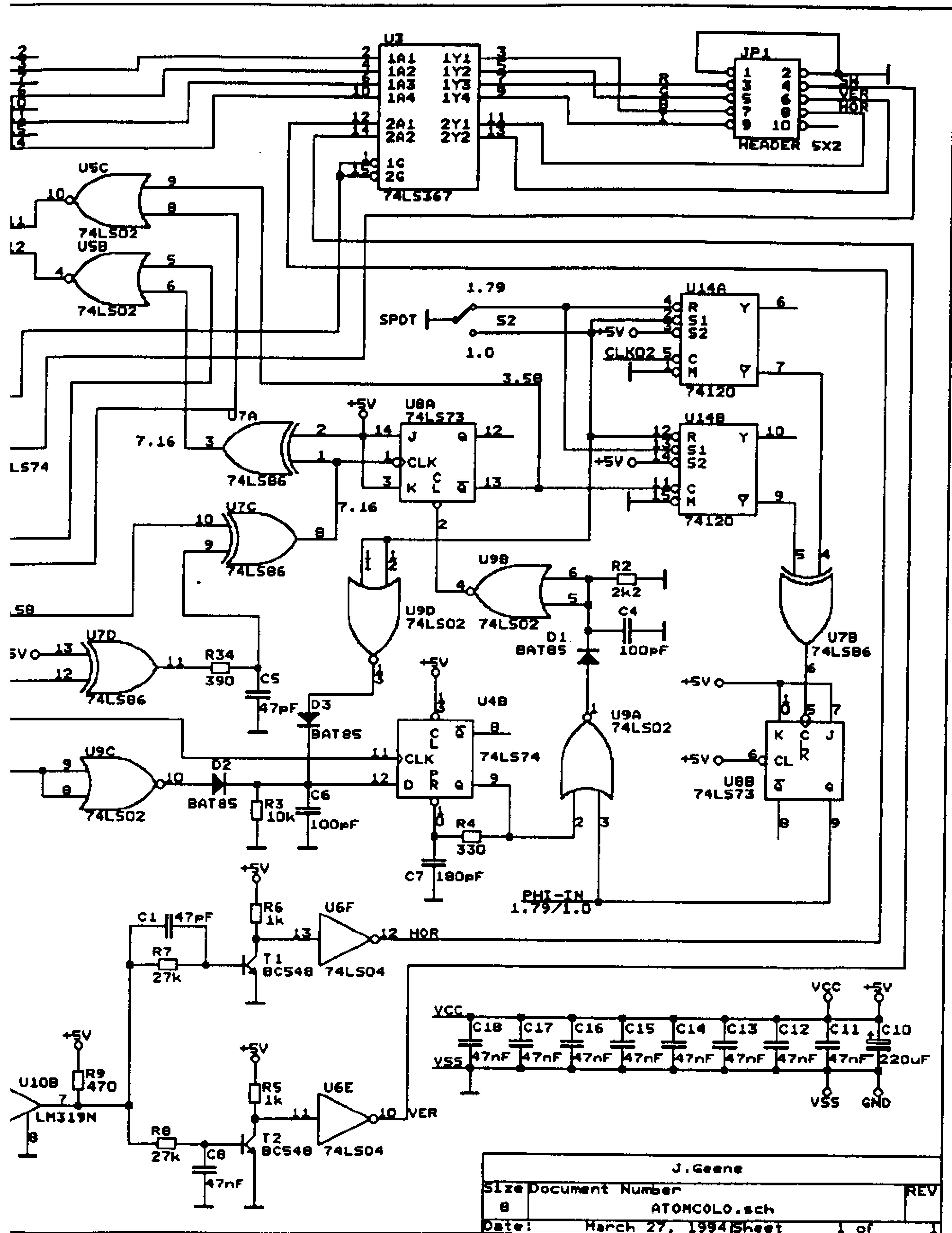
Kleur	Adres CSS 765	YABCD 43210	Adres #	Data IRGB	#
1) ATOM mode.					
RED	000	11101	1D	0100	4
GREEN	000	10000	10	0010	2
BLUE	000	10111	17	0001	1
YELLOW	000	10100	14	0110	6
ORANGE	001	11101	3D	0100	4
BUFF	001	10000	30	0111	7
MAGENTA	001	10111	37	0101	5
CYAN	001	10100	34	0011	3
2) CGA mode 1.					
RED	100	11101	9D	0100	4
GREEN	100	10000	90	0010	2
BLUE	100	10111	97	0001	1
YELLOW	100	10100	94	1110	E
ORANGE	101	11101	BD	0110	6
BUFF	101	10000	B0	0111	7
MAGENTA	101	10111	B7	0101	5
CYAN	101	10100	B4	0011	3
3) CGA mode 2.					
RED	110	11101	DD	0100	4
GREEN	110	10000	D0	0010	2
BLUE	110	10111	D7	0001	1
YELLOW	110	10100	D4	1110	E
ORANGE	111	11101	FD	1100	C
BUFF	111	10000	F0	0111	7
MAGENTA	111	10111	F7	0101	5
CYAN	111	10100	F4	0011	3
4) ATOM mode 2.					
RED	010	11100	5C	0100	4
GREEN	010	10101	55	0010	2
BLUE	010	11111	5F	0001	1
YELLOW	010	10001	51	0110	6
ORANGE	011	11100	7C	0100	4
BUFF	011	10101	75	0111	7
MAGENTA	011	11111	7F	0101	5
CYAN	011	10001	71	0011	3

Verklaring van de modes, welke instelbaar zijn met S1. S1 stelt A6 en A7 in, corresponderend met de kolommen 6 en 7 in tabel 1.

ATOM mode 1: Deze mode geeft dezelfde codering als de originele ATOM kleurenkaart. Op een CGA monitor geeft geel dan bruin.
CGA mode 1: Geel is hier geel en oranje is hier bruin.



U12A
U12B
U11A
U11B
U10A
LM319N
U1
82S129
U2
74LS379
U5D
74LS02
S1
SW DIP-2
U6C
74LS04
U4A
U6D
U5A
U6A
U13
6847
JP3
HEADER 20
JP2
U6B
T3
BC558
T4
BC548
T5
PC3
PC0



U3
74LS367
JP1
HEADER 5X2
SH
VER
HOR
U5C
74LS02
U5B
74LS02
SPDT
1.79
S2
1.0
U14A
74120
CLKO2
3.58
U14B
74120
+5V
U8A
74LS73
U7A
74LS86
7.16
LS74
U7C
74LS86
U9D
74LS02
U9B
74LS02
R2
2k2
C4
100pF
D1
BAT85
U7B
74LS86
U7D
74LS86
R34
390
C5
47pF
D3
BAT85
U4B
74LS74
U9A
74LS02
U8B
74LS73
U9C
74LS02
D2
BAT85
R3
10k
C6
100pF
R4
330
C7
180pF
PHI-IN
1.79/1.0
C1
47pF
R7
27k
R6
1k
T1
BC548
U6F
74LS04
HOR
VCC
VSS
C18
C17
C16
C15
C14
C13
C12
C11
C10
47nF
220uF
GND
R9
470
U10B
LM319N
R5
1k
T2
BC548
U6E
74LS04
VER
R8
27k
C8
47nF
J.Geene
Size
Document Number
REV
B
ATOMCOLO.sch
Date:
March 27, 1994
Sheet
1 of 1

CGA mode 2: Oranje is hier licht rood, verder gelijk aan mode 1.
ATOM mode 2: Is alleen van toepassing als pin 39 (CSS) van de 6847 1 is. Alpha karakters zijn dan rood i.p.v. groen.
Als een CGA kleurenmonitor gebruikt wordt is CGA mode 2 te prefereren, bij een SCART monitor kies dan ATOM mode 1.

d) RGBI latch.

Daar de data op de uitgangen van de PROM niet precies tegelijk aankomt en dus niet stabiel is gedurende de gehele pixeltijd, wordt de latch U2 gebruikt om de RGBI signalen te klokken met de pixelfrequentie. Daar de monochrome pixeltijd gelijk is aan de halve kloktijd is de pixelfrequentie dus 7.16 MHz. Daar de horizontale resolutie bij kleur halveert, is de frequentie dan 3.58 MHz. De omschakeling wordt verzorgd door U5B, U5C en U5D.

e) Sync scheiders.

Met comparator U10B wordt het composiet sync signaal gescheiden van het op de ingang aangeboden video signaal (Y signaal). Met R1 wordt de drempel ingesteld. Hierna een integrerend netwerk R8/C8 om de vertikale sync-puls er uit te halen en een differentierend netwerk C1/R7 om de horizontale sync-puls er uit te halen. De pulsen worden geinverteerd en gebufferd door U6F en U6E. Voor een CGA monitor zijn gescheiden sync-pulsen noodzakelijk.

f) De pixelklok.

Uitgaande van de klokfrequentie op pin 33 van de 6847 van 3.58 MHz wordt door U7D, R34, C5 en U7C de frequentie verdubbeld tot 7.16 MHz. Dit is een bekende schakeling voor dit doel. De verdubbelde frequentie wordt aan de omschakelaar en aan flipflop U8A toegevoerd zodat ook weer de 3.58 MHz op de omschakelaar ter beschikking staat. U8A heeft ook nog een andere functie, zie verder hieronder bij h).

g) De omschakeling met poorten PC0 en PC3.

De werking van PC3 is hetzelfde gebleven als in de originele ATOM. PC3 zat verbonden met CSS van de 6847 en daarmee kon de kleurenset omgeschakeld worden. Bij monochrome monitors veranderde dan de helderheid. In deze schakeling wordt PC3 aangesloten op A5 van de PROM U1. De RGBI codering wordt met A5 naar een andere PROM inhoud geschakeld. Zie tabel 1, kolom 5. Deze manier van schakelen werkt veel beter dan op de oude manier (?#B002=?#B002:8).
PC0 is een van de poorten die gebruikt worden voor de cassette interface, maar kan vrij gebruikt worden voor andere doeleinden als deze interface niet gebruikt wordt (Zie ook AT&P blz. 194).

In de nieuwe kleurenkaart wordt PC0 gebruikt om te schakelen van ATOM mode naar 80 koloms mode. Bij inschakelen van de computer wordt PC0 op 1 gezet en komt deze als 0 op de D-ingang van U4A. Deze 0 wordt door het FS signaal naar de Q uitgang geklokt en daardoor wordt buffer U3 geactiveerd en staat de computer in ATOM mode. Wel moet dan de clearingang van U4A hoog zijn. Deze is hoog als de 6847 in tekst mode staat. In graphics mode is A/G hoog en dus de clearingang van U4A laag zodat in graphics mode (clear 1 t/m 4) niet naar 80 kolom mode geschakeld kan worden. De Q-not uitgang van U4A schakelt tegelijkertijd via connector JP1 de buffer op de 80 kolom kaart uit. Het zal niet moeilijk zijn om te zien dat als PC0 0 wordt (?#B002=?#B002:1) dat dan van ATOM mode naar 80 kolom mode geschakeld

wordt. Op de een van de vorige bladzijden is het al even aan de orde geweest: de horizontale resolutie is 256 pixels in CLEAR 4 en in ALPHA mode en 128 pixels in alle andere graphics modes, inclusief de colour modes. De latch U2 moet eenmaal per pixel geklokt worden, dit houdt in dat de klokfrequentie voor 256 pixels 7.16 MHz en voor 128 pixels 3.58 MHz moet zijn. De signalen GM0 en A/G zorgen samen met poort U5A en U6D dat deze klokfrequentie alleen naar de lage frequentie geschakeld wordt als de 6847 in COLOUR mode en in GRAPHICS mode staat. In alle andere situaties is de klokfrequentie van U2 7.16 MHz.

Het blijkt dat in CLEAR 1 t/m 3, met een resolutie van 128 pixels in monochroom mode, het niet nodig is dat met de lage frequentie geklokt wordt. In COLOUR mode is dat beslist nodig, omdat anders de vertikale lijnen dikker worden dan ze zijn moeten.

h) Omschakeling 1.0 naar 1.79 MHz.

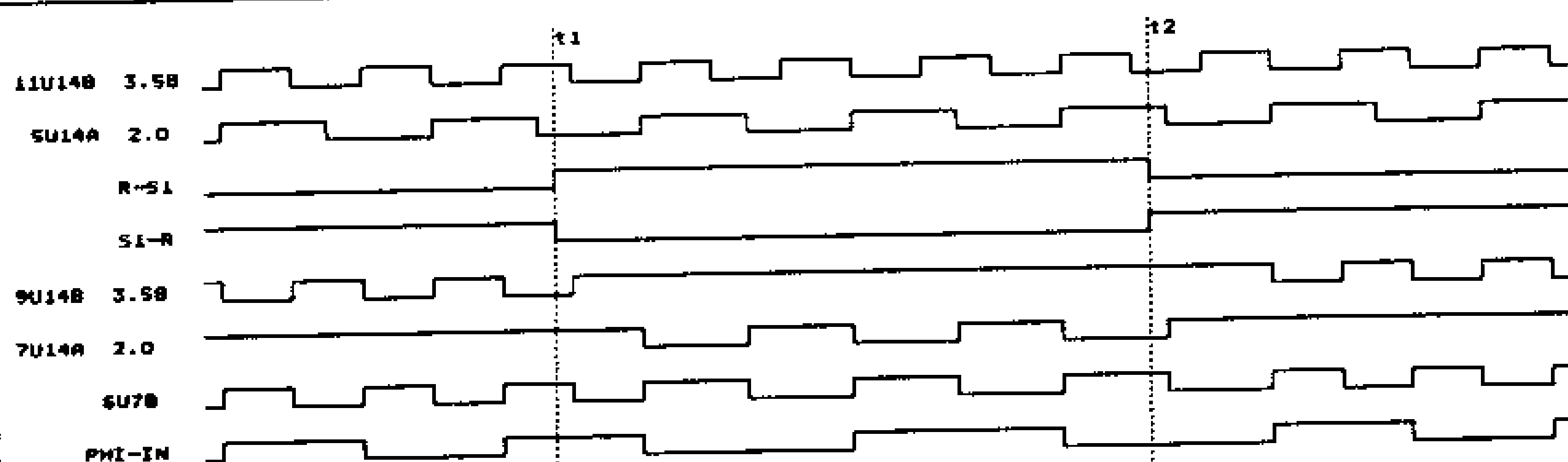
De bekende schakeling met 2 (74LS73) flipflops om van 1.0 naar 2.0 MHz om te schakelen werkt hier niet omdat de signalen niet aan elkaar gekoppeld zijn. Ze komen n.l. van 2 geheel verschillende klokgeneratoren. Ik heb lang gezocht naar een oplossing en heb die uiteindelijk gevonden in een 74120 (Dual pulse synchronizers/drivers) waar J. Swinkels in A.N. 9-3 blz. 10 een uitleg van de werking heeft gegeven. Het zijn in feite 2 R-S flipflops U14A en U14B, die na omschakelen het kloksignaal pas doorgeven op de eerstvolgende opgaande flank van het kloksignaal. Het stoppen gebeurt direct na het afschakelen, zodat geen te korte klokpulsen ontstaan en de 6502 processor niet gaat hangen.

Bestudeer de figuur "Frequentie 1.0 - 1.79 MHz" en het artikel van J. Swinkels om een en ander te kunnen volgen. Deze schakeling heeft dezelfde functie als de poorten 1, 2 en 3 in het autosync schema van P. Ehrlich.

i) Autosync 1.79 MHz.

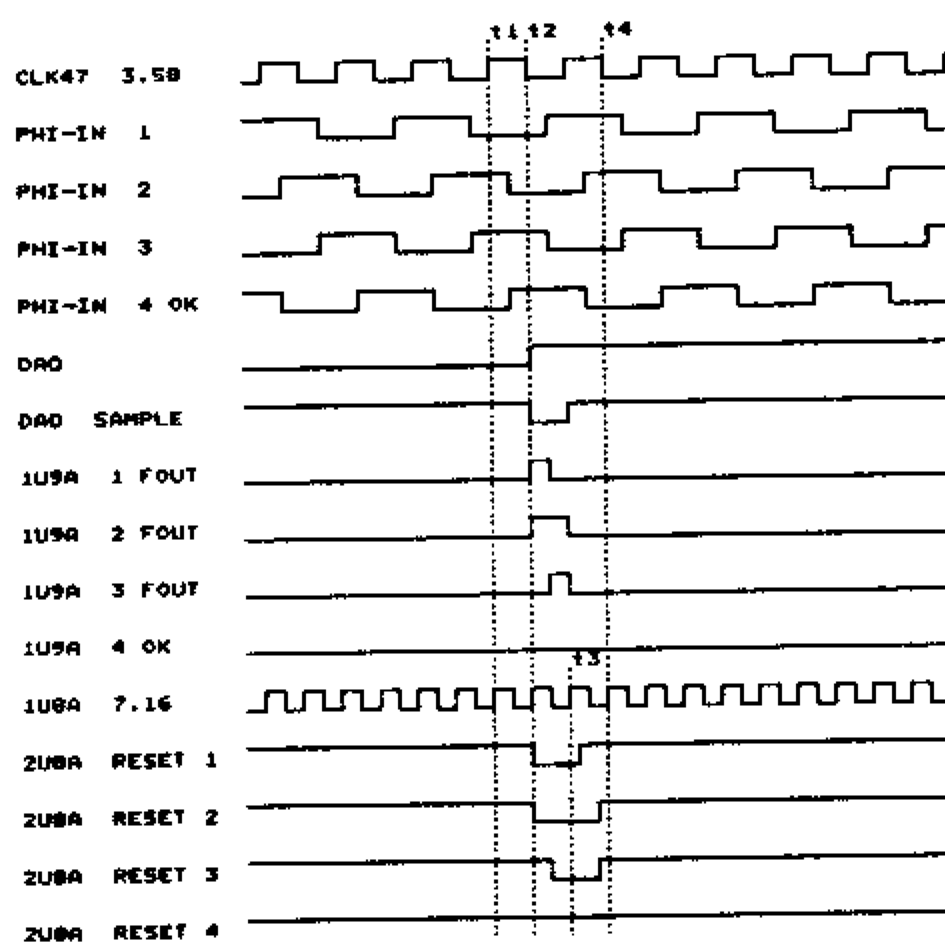
De nu volgende beschrijving van de autosync 1.79 (1.8 voor de afronders) is alleen de andere uitvoering van de schakeling. De principiele werking heeft P. Ehrlich in A.N. 5-1 blz. 20 perfect uitgelegd, dat ga ik dus niet nog eens doen. Het is natuurlijk erg handig om dit artikel erbij te nemen als men geinteresseerd is in de werking. Hierboven is de wissel al besproken. Ook het omzetten van de klokfrequentie van 3.58 naar 7.16 MHz is al vermeld.

In het timingdiagram "Autosync 1.79 MHz" is begonnen met de faserelatie van PHI-IN en DA0. Vier verschillende toestanden zijn mogelijk, waarbij alleen toestand 4 ok is. In flipflop U4B wordt een sample gemaakt van DA0 (DA0 sample). De tijdsduur moet ongeveer 140 nS zijn en wordt bepaald door R4/C7. Dit sample wordt in U9A vergeleken met het PHI-IN signaal. Dit is het kloksignaal van de 6502. Afhankelijk van de fase van PHI-IN ontstaat op 1U9A een van de drie fout signalen of het ok-signaal. Bij dit ok-signaal gebeurt er verder niets, maar bij een van de fout signalen wordt aan flipflop U8A een resetpuls via U9B toegevoerd. Met behulp van D1/R2/C4 is de foutpuls op 1U9A zodanig verlengd dat altijd 1 periode van het 7.16 MHz signaal wordt geblokkeerd. Zie tijdstip t3. De resetpuls is 0 tijdens de opgaande flank van het 7.16 MHz signaal. Dit proces herhaalt zich eventueel totdat de faserelatie PHI-IN en DA0 ok is. In het timingdiagram zien we dat in dat geval er geen resetpuls ontstaat. Poort U9C/D2/R3/C6 blokkeert het vergelijken als de 6847 actief wordt. Poort U9D/D3 blokkeert het vergelijken in de 1.0 MHz stand. Als deze faserelatie ok is, dan is RUISVRIJ plotten een feit.



FREQUENTIE 1.Ø – 1.79 MHz

Size	Document Number	REV
A		
Date: December 4, 1994	Sheet of	



AUTOSYNC 1.79 MHz

Size	Document Number	REV
A		
Date: December 4, 1994	Sheet of	

j) De signaalbuffer

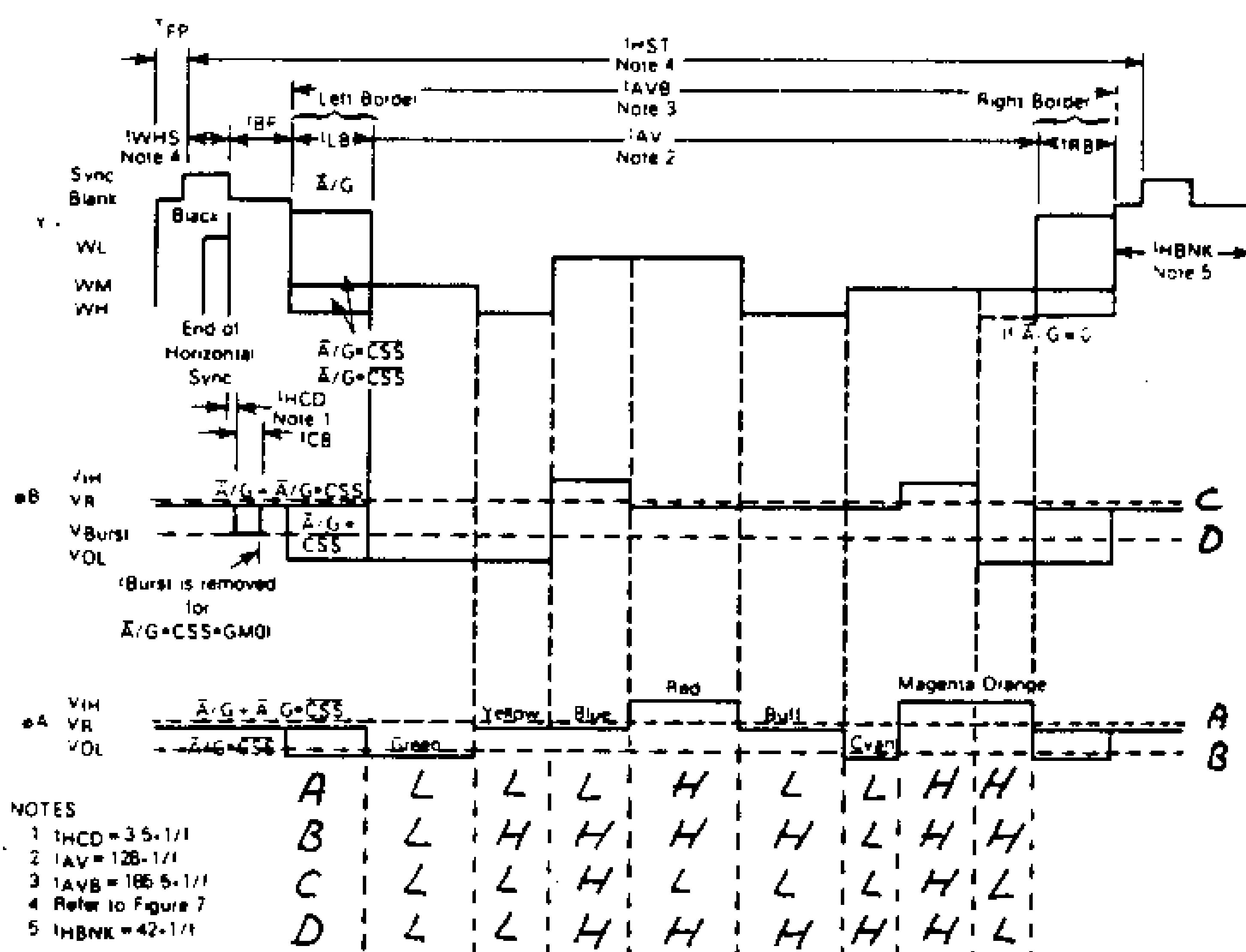
De buffer U3 buffert de videosignalen en de sync signalen om die over een langere afstand te kunnen transporteren. De buffer heeft een tri-state uitgang zodat de omschakeling van ATOM-mode naar 80 kolom-mode gemakkelijk kan gedaan worden. De buffers in de beide kaarten staan om beurten aan. De uitgangen kunnen dus aan elkaar verbonden worden via JP1.

De opbouw van de kaart is bij mij als volgt: ik heb de originele ATOM kaart doorgezaagd zodanig dat de LM319's en de 6847 er nog op zaten, daarna heb ik er een gaatjes print aangebouwd waar de rest van de spullen op staan. De totale print is dan weer ongeveer net zo groot als voorheen. JP2 en JP3 zijn connectors op de plaats van de ic-voet op de ATOM hoofdprint. De signalen PC0, PHI-IN, CLK02 en PC3 worden via deze connectoren geleidt, zodat de kleurenkaart meteen aangesloten is. Deze pennen op JP2 en JP3 moeten wel vrijgemaakt c.q. aangesloten worden. Alleen een bandkabel van JP1 naar de color/colour interface is nodig. Het PHI-IN signaal komt direct aan pin 37 van de 6502. Het CLK02 (2 MHz) signaal komt van pin 11 van IC44.
Bij eventuele belangstelling ben ik bereid een printontwerp te maken.

Literatuur:
A.N. 10-2 deel 1.
A.N. 12-3 deel 2.
A.N. 9-3 ic 74120.
A.N. 5-1 autosync 1.8.
Motorola databoek.
Atom colour board contents.

Sjaak Geene,
Zonneweide 6,
5221 BH 's Bosch.
Tel. 073 312080.

FIGURE 10 – VIDEO AND CHROMINANCE OUTPUT WAVEFORM RELATIONSHIPS



Atomisme.

In de laatsts Atomnieuws werd gevraagd om copy en/of ideeen. Nu ligt er bij mij al geruime tijd een idee op de plank, waaraan ik nu en dan stap voor stap een stukje verder werk. Het gaat om het volgende:
Een echte PC heb ik nog steeds niet en ik zie er ook eigenlijk nog steeds niet helemaal het nut van in. PC's hangen ook met de regelmaat van een klok, alleen als m'n Atom hangt is er meestal weinig aan de hand en is bijna altijd het bestand te redden. Nog steeds doet m'n Atompje redelijk trouw wat ik van hem verlang. M'n (simpele) boekhoudinkje loopt als een trein. Je kunt er een brief of wat dan ook netjes op uittikken zonder dat je, zoals bijv bij WP 5.zusemmezo, na verloop van tijd met een verwilderde blik in de ogen aanbelt bij een psychiatrische inrichting of je er alstublieft in mag.
Al een hele tijd vind ik m'n Atom met al z'n uitbreidingen een nogal omvangrijk apparaat geworden. Het lijkt me een stuk makkelijker als ik alleen m'n Atomkast en een monitortje op m'n bureau zou hebben staan. Een soort Atom Notebook zogezegd.
Zodoende dus het idee de "standaard" Atom, met al z'n oorspronkelijke mogelijkheden op een dubbelformaat eurokaart te proppen. Volgens mij zou dat moeten kunnen met de tegenwoordige componenten. De C,D,E en F ROM kunnen in 1 Eprom bijvoorbeeld. Het hele lage geheugen past in een IC. Voor de schakelkaart denk ik er aan het systeem dat Willem Kautz een paar jaar gelden ontwikkelde en dat uitstekend werkte te integreren. Ook een voeding voor het geheel zal dan niet al te zwaar hoeven te zijn, de tijd dat de tram langzamer ging rijden als er iemand z'n Atom aanzette is toch wel voorbij dacht ik zo. M'n gedachten gaan dan uit naar een Atomkast met daarin:

De nieuwe dubbel eurokaartformaat Atom
Gesandwitched met de CP/M kaart
Een plekje voor de 80 kolomskaart
Een simpele lineaire voeding met zo'n plat rond trafootje
Zo'n klein 3.5 inch driveje

Het ligt in de bedoeling om de printer- en de 64 pins uitgang aan de achterzijde op deze plaats te handhaven, teneinde toch het gebruik van uitbreidingen tot de mogelijkheden te blijven laten behoren.
U ziet, ik heb een hoop noten op mijn zang. Helaas heb ik de laatse tijd bar weinig tijd wegens studie en deze situatie zal zich nog wel even voortzetten. Als ik tijd van leven heb hoop ik dit toch wel voor elkaar krijgen

Theo Waayer

```
                                    II

UU      UU      NN NNNNN       II     CCCCCCC       OOOOOO      RR RRRR     NN NNNNN
UU      UU      NNN      NN    II    CC       C    OO      OO   RRR     RR  NNN      NN
UU      UU      NN       NN    II    CC            OO      OO   RR      RR  NN       NN
UU      UU      NN       NN    II    CC            OO      OO   RR          NN       NN
UU      UU      NN       NN    II    CC            OO      OO   RR          NN       NN
UU      UU      NN       NN    II    CC            OO      OO   RR          NN       NN
UU      UU      NN       NN    II    CC       C    OO      OO   RR          NN       NN
  UUUUUU        NN       NN    II     CCCCCC        OOOOOO      RR          NN       NN

BBBBBB    BBBBBB    SSSSSS          ONLINE:      00:00-24:00 Hour/Day
BB   BB   BB   BB   SS                SysOp:     Henri Derksen
BB   BB   BB   BB   SS              CoSysOp:     Bob Brand
BBBBBB    BBBBBB    SSSSSS          AcoNetNode:  77:8500/504
BB   BB   BB   BB       SS          Telephone:   +31 (0)85-425506
BB   BB   BB   BB       SS           C.A.T.:     P.O.Box 1006, 6801 BA  ARNHEM
BBBBBBB   BBBBBBB   SSSSSS                       Gelderland    The Netherlands
```

Wat is een BBS.

De afkorting staat voor: Bulletin Board Systeem. Ofwel een soort prikboard om berichtjes in achter te laten (net zoals bij Albert Hein e.d.).
Het heeft een Computer en een modem nodig. Als er iemand belt zorgt de software ervoor dat het modem de telefoonlijn opneemt, en dat het BBS aan de gebruiker gepresenteert wordt.
Wat krijg je zoal te zien, en wat is de logische indeling:

CONNECT XXXXX Je hebt een digitale verbinding tot stand gebracht.

Press ESCAPE twice to enter the BBS, of wacht 11 seconden
Dit is nodig om te testen of er een Mailer belt. B.v. een collega BBS of Point.

Dan krijg je het inlogscherm te zien, waarna om je voornaam en achternaam gevraagd wordt, als,ede je wachtwoord als dat al bekend is. Zo niet dan wordt je als een nieuwe user aangemerkt en moet je een paar simpele vragen beantwoorden. Een volgende keer ben je dan wel bekend.

Na het inlogscherm, komt er een Bulletins menu. Hierin staan een aantal teksten in algemene zin. B.v. Computer Club Activiteiten, Antivirussen etc.

Hieronder een voorbeeld van een inlogsessie:

[Af en toe is er door de redactie commentaar toegevoegd en wat ingekort.]

```
*****************************************************************

ATZ
OK
ATDT 085 425506
```

```
CARRIER 14400                 \
PROTOCOL: LAP-M                \  Meldingen van het Modem
COMPRESSION: V.42BIS           /
CONNECT 57600/ARQ             /

FrontDoor 2.02; Noncommercial version

Press Escape twice for UniCorn

U heeft nu aan de Lijn:

Naam:       UniCorn BBS  (085-425506).
Sysop:      Henri Derksen.
Doel:       Berichten en Files uitwisselen.
Points:     DownLinks zijn welcome !
Open:       24 uur per dag, 7 dagen per week.

Modem:      Micro Technology MT3242.
Snelheden: V21, V22, V22bis, V23, V32, V32bis
            300, 1200, 2400, 4800, 7200, 9600, 12000, 14400
Bits Per Seconde
   Inclusief 1200/75 Bps !  Full Duplex  8N1 ANSI/TTY  BBS
   ErrorCorrectie: V42 LAPM en zonodig FallBack naar MNP 1-4,
   DataCompressie: V42bis   en zonodig FallBack naar MNP 5.

Node:       77:8500/504    @ AcoNet
AKA         2:2801/208     @ FidoNet
AKA         220:800/208    @ TotaalNet
AKA         85:100/310     @ 085-Net

Momentje,  Even omschakelen van de FrontDoor Mailer naar
RemoteAccess BBS, dat swappen duurt wel even, geduld A.U.B.

      [ Er volgt een welkoms-scherm met enkele mededelingen]

You will be prompted for your name and password now, please
use your REAL name.

Next system event at ^V (local time), in   ^U minutes.

Please enter your Full Name: Henri Derksen

Scanning user-file ...

Password: *******
NOTE: This system will be unavailable in  204 minutes!
Chatten (Yellen) alleen tussen 20:00 en 23:00 Uur !
Security status:  Sysop

You have Sysop access to this BBS. Please enjoy your stay.


Check for waiting mail (Y/n)? No
```

Check for new files (y/N)? No

```
******************** News van de Dag ****************************************

Hallo Allemaal, Nieuwe UniCorn BBS-gebruikers,

Nu eindelijk de ruimte: 700 MB OnLine en OOK op Hoge snelheid !

Dit BBS is UNIek want het is het enige BBS dat echt Alle aCORNs ondersteunt
Ook MS-Dos PC en andere (oude) HuisComputeraars zijn welkom !

UniCorn BBS brengt u het nieuws op de juiste SNELheid !    kijk maar:

Nu al op 300/300, 1200/75, 1200/1200 en 2400/2400 Bits Per Seconde.
Sinds 1 Maart 1992 ook nog op V32 9600 BPS en V32bis 14.400 BPS ?
Leuk voor points en andere pollers !

Lees ook even de BULLETINs

UniCorn BBS  Daar kun je niet omheen als ACORNist !
```

```
       (     )                                        (c) C.A.T. 1992
     (   )
    ( )
   |   |
    \ /     ____
    | |_____|  |\
  | UniCorn BBS |  |B||U||L||L||E||T||I||N||S|   Pulling  PC's
  |__/\______/\__|-                              All Hobby Computers
     <>      <>       <>       <>       <>       <>
----------------------------------------------------------------------------
   0 = SysteemInfo                         3 = Info Acorn Big Ben Club
   1 = UniCorn BBS                         4 = Info Acorn Atom Club
   2 = New User Info                       5 = Info AcoNet
   6 = Net Mail berichten versturen       10 = Acorn Big Ben Club Regio Oost
   7 = DataCommunicatieProtocollen             (Agenda 1994)
   8 = Organisatie Software               11 = Acorn Atom Club Regio Arnhem
   9 = Anti4ussen !                            (Agenda 1994)
  13 = Veel gestelde Acorn vragen FAQs    12 = Info Auditief Gehandicapten

File to view (Enter=quit):
```

[U kunt door een van bovenstaande items te kiezen informatie lezen over het betreffende onderwerp]

```
  UniCorn BBS

     ____
    /    \       /====...=====/
   ( @  @ )     /  Main menu /
    \____/     /=====...====/
=)\   |       /
   \-O----/
    /  \
```

```
Message section      File section         Goodbye (logoff)   Statistics
Bulletin info        Yell the Sysop       Change Setup         User list/search
     Extra info           Version              Todays callers   Hours of usage
  Outside               *ysOp
Select: M
```

```
 ____________________________
|\   The   MessageBase      /|
| \  From UniCorn BBS     /  |
|  \_________     _______/   |
|            \/              |
|        Read it             |
|____________________________|
```

```
There are 74 messages in this area.
System contains  4746 messages, ranging  1228 to  27529.

Msg.area #   3 ...  SysOp's                             U

Area change          Read (read msg)      Enter message       List (brief)
Check mail-box       Verbose msg list     sWich Combined Area's On/Off
Do Read Combined     Statistics           MAIN MENU           Net Adressen Lijst
             Browse nodelist       Goodbye (logoff)
Select: A
```

```
  1 ... Algemeen binnen UniCorn       U     2 ... Handel  (FreeLink & AcoNet) A
  3 ... SysOp's                       U     4 ... Klets/Onzin  (AcoNet)       A
  5 ... DataCommunicatie              A     6 ... Acorn Big Ben Club          A
  7 ... AcoNet Algemeen               A     8 ... AcoSysop's                  A
  9 ... Net-Mail (van BBS naar BBS)N       10 ... AcoNet Bestuur              A
 11 ... 27 MHz Bakkies                T    12 ... Auto's                      T
 13 ... Acorn Atom                    U    14 ... Atari Computer(s) TotaalNet T
 15 ... SysOp Support                 T    16 ... Hackers Info TotaalNet      T
 17 ... Argus & Stentor Modem         F    18 ... Boeken   (AcoNet)           A
 19 ... Test   ACOnet                 A    20 ... BBS&Mailer Support (AcoNet) A
 21 ... DataCommunicatie TotaalNet    T    22 ... VraagBaak   Acorn's         A
 23 ... Programming AcoNet            A    24 ... HardWare  Acorn's           A
 25 ... HardWare TotaalNet            T    26 ... International BBC           F
 27 ... International Archimedes      F    28 ... International Acorn Mailers F
 29 ... Auditief Gehandicapten        U    30 ... Knooppunt Arnhem/Nijmegen   M
 31 ... Operating Systemen (TN)       T    32 ... SysOp's 085-Net             M
 33 ... Handel TotaalNet              T    34 ... Muziek  (AcoNet)            A
 35 ... EcoNet Local Area NetwerkenA       36 ... RadioZendAmateurs Totaalnet T
 37 ... BBSinfo (Reclame SysOp's)     A    38 ... TotaalNet SysOp's           T
 39 ... Remote Access Support (TN)    T    40 ... TotaalNet BabbelBox         T
 41 ... Sex Algemeen                  T    42 ... MutiMedia TotaalNet         T
 43 ... MultiTasking TotaalNet        T    44 ... NetZaken Totaalnet          T
 45 ... Chat 085-Net                  M    46 ... Algemeen TotaalNet          T
 47 ... International Acorn SysOp'sF       48 ... ProgrammeerTalen (TN)       T
 49 ... Handel 085-Net                M    50 ... Lokaal_Nieuws-085-Net       M
 51 ... Virus Forum TotaalNet         T    52 ... UniCorn DownLinks (Pollers) U
 53 ... FileZoekers FreeLink          A    54 ... Virus Informatie TotaalNet  T
 55 ... VraagBaak  TotaalNet          T    56 ... Mailers algemeen (FreeLink) T
 57 ... Modems (FreeLink)             T    58 ... BBSsen algemeen (FreeLink)  T
 59 ... TotaalNet Hub 20              T    60 ... FreeLink Chat/Klets         A
 61 ... Archimedes BinkleyTerm        F    62 ... Achimedes Communications    F
 63 ... Archimedes Programming        F    64 ... UseNet                      A
 65 ... Moppen 085-Net                M    66 ... Mededelingen & vragen AcoNetA
 68 ... 2801                          F    69 ... AcoNet SoftWare OntwikkelingA
 70 ... Disable.512                   F    71 ... Geanealogie                 A
 73 ... COMP.SYS.ACORN.ANNOUNCE       A    74 ... COMP.SYS.ACORN              A
 75 ... COMP.SYS.ACORN.TECH           A    76 ... COMP.SYS.ACORN.ADVOCACY     A
 77 ... Gehandicapten 028             F    78 ... EcoNet (International)      F
 79 ... Puzzel (AcoNet)               A    80 ... Big Ben Club Regio Oost     U
 81 ... Bestuur Big Ben Regio Oost U       82 ... Comp.Sys.Acorn.Games        A
```

```
 83 ... AcoNet Hub 5 DownLinks      A    84 ... Comp.Binaries.Acorn        A
 85 ... Comp.Sources.Acorn          A    86 ... ACO.INT.BBS&Mailer         A
 87 ... ACO.INT.DataComm            A    88 ... ACO.INT.General            A
 89 ... ACO.INT.HardWare            A    90 ... ACO.INT.Questions          A
198 ... Bad Messages                    199 ... Dupe Messages
Select area: 13
```

[Door hier nu een gebied te kiezen kunt u berichten lezen en verzenden in dit berichten gebied. Het is de bedoeling dat u zich houdt aan het onderwerp, dus geen Moppen in de Handel area.]

```
There are 1 messages in this area.
System contains  4746 messages, ranging  1228 to  27529.

Msg.area #   13 ...  Acorn Atom                    U

Area change          Read (read msg)      Enter message        List (brief)
Check mail-box       Verbose msg list     sWich Combined Area's On/Off
Do Read Combined     Statistics           MAIN MENU            Net Adressen Lijst
             Browse nodelist      Goodbye (logoff)
Select: R

(F)orward,  (R)everse,  (I)ndividual, (H)elp,
(M)arked,   (N)ew msgs, (S)elected, (Q)uit.
Select: Reverse
Message area "Acorn Atom                    U" contains 1 messages.
System contains messages ranging 1228-27529.
Enter message number to start at (Enter=First/Last):
Pause after each message (Y/n)? Yes

Message #9490 - Acorn Atom                    U
   Date: 14-09-94 16:07
   From: Henri Derksen
     To: All
Subject: DownLoad gevuld

Hallo Atom liefhebbers,

Door de prettige medewerking van Roland Leurs heb ik de DownLoad voor onze
Atom kunnen vullen met de meest recente ATOM in PC software.
En uiteraard nog wat ander leuk spul.

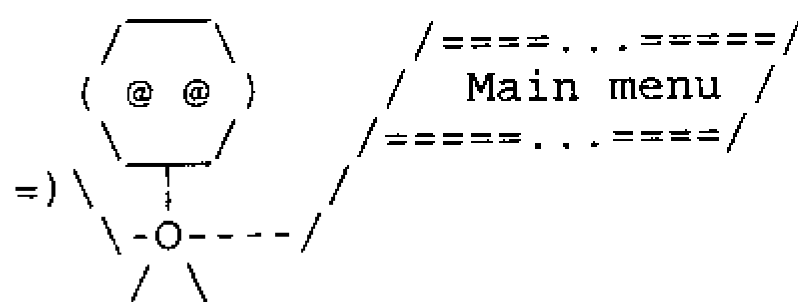
Doe er je voordeel mee
-+-
 # Origin: Connectivity is the Future; UniCorn BBS 31 85 425506 (77:8500/504)


(A)gain, (N)ext, (L)ast, (R)eply, (E)nter, (D)elete, (!*X/=), (S)top: Next

End of messages
Press (Enter) to continue:
```

[Van hieruit maken we een sprong naar het hoofdmenu.]

UniCorn BBS

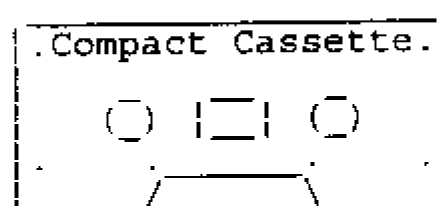


Message section
Bulletin info
Extra info
Outside
File section
Yell the Sysop
Version
*ysOp
Goodbye (logoff)
Change Setup
Todays callers
Statistics
User list/search
Hours of usage
Select: F

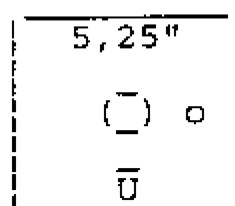
UniCorn BBS (c) C.A.T. 1991

File-Section and Transfer:

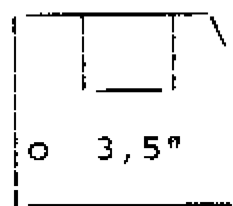
From BBSsytem via Wire 2 Your Rotating Memory
The SoftLoverToys:



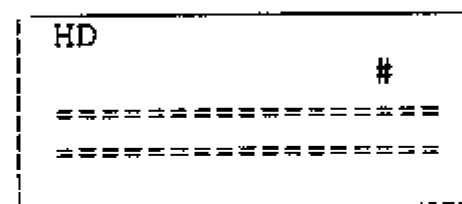
Slow Love



Fast Love



Metal Love



Instant Love.

Ga met AreaChange naar de Juist FileArea om te DownLoaden !

UpLoads komen altijd in FileArea 17, wel zichtbaar, niet te DownLoaden.

File Area # 21 ... Archivers
Area change
Upload (send)
Wild-card search
DownLoad for SysOps
Locate
Download (receive)
Raw dir
Outside
Files list
Statistics
Copy File > andere Area
Goodbye
MAIN MENU
Select: A

File Areas: ----------

1 ... Algemeen binnen UniCorn BBS
2 ... Sysop's
3 ... DataCommunicatie
4 ... Acorn Archimedes
5 ... Acorn BBC/Master/Electron
6 ... Acorn Atom
7 ... MS-Dos (Andersdenkenden)
8 ... VierRussen
9 ... Utils Heel Handig
10 ... Argus & Stentor Modem
11 ... Cambridge Z88
12 ... Mailer Support
13 ... BBS Support
14 ... Unix
15 ... CP/M
16 ... 32016
17 ... UpLoads
18 ... BasiCode
19 ... TeleText
20 ... Electronische Tijdschriften
21 ... Archivers
22 ... ProgrameerTaal C
23 ... ProgrameerTaal Pascal
24 ... Auditief Gehandicapten
25 ... FSC Fido Standard DoCuments
26 ... FTS Fido Technical Standard
27 ... Demo's
28 ... ACO-FAQS
29 ... ACO-File A
30 ... ACO-BBCSCAN
31 ... Big Ben Club
32 ... Aco-FrontDoor
33 ... ACO-INFO A
34 ... Aco-SoftWare
35 ... ACO-WimpLink A
36 ... Packet Radio
37 ... VirusInfo T
38 ... Imail (MailTosser/Scanner)
39 ... Handicap
40 ... TotaalNet NodeLijsten

```
41 ... FreeLink NodeLijsten         A   42 ... FreeLink Node Diffs
Select area: 6

File Area #   6 ... Acorn Atom
Area change          Locate                Files list          Goodbye
Upload (send)        Download (receive)    Statistics          MAIN MENU
Wild-card search     Raw dir               Copy File > andere Area
DowNload for SysOps Outside
Select: F


Press P to Pause, S to Stop
+--------------------------

 vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
 UniCorn BBS  FileArea:  6  Acorn Atom

 Files voor de Acorn Atom
 +-----------------------------------------------

AN-TRACE.ZIP    22263 16-08-94  Acorn Atom Nieuws Item tracer, bijgewerkt t/m
                                jaargang 12 (1993).
ATOM_220.ZIP   202740 05-06-94  Atom operating soft, incl terminalprogramma
                                v2.20 voor Acorn Atom in PC kaart.
ATOMLIST.ZIP     8666 16-08-94 Laat de inhoud van een Atom Basic file zien
                                vanuit DOS
ATOMERR.INC      6469 03-12-94 *Include file for ASM atomPC terminal 2.2
DARNIT.RUN       5881 20-11-93 Darnit, spel voor Atom-in-pc, ready-to-run
FORTUIN.ZIP     15884 16-08-94 Rad van fortuin, spel voor Atom-in-pc
FRONT.ARC      201021 20-09-91 Acorn Atom plaatjes van de Atom Nieuws Cover,
                                geARCed met Tekening.EXE voor de PC
PIC-WPG.ARC    184051 01-01-90 Acorn Atom Plaatjes gecomprimeerd.
PLAAT2.ARC     119832 29-12-92 Vervolg Atom Plaatjes, zie Atom Nieuws 1993
                                No.1 Pagina 4.
REVERSI.RUN      2566 08-11-93 Spel voor Atom-in-pc, ready-to-run
ROTFAN.RUN       1291 16-08-94 Grafische demo voor Atom-in-pc, ready-to-run
SNAPPER.RUN      5371 12-10-93 Spel voor Atom-in-pc, ready-to-run
TEKENING.EXE    41957 29-12-92 PC-Programma om *.AGS Acorn Atom plaatjes te
                                tonen.

 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 UniCorn BBS  FileArea:  6  Acorn Atom
 +-----------------------------------------------

Press (Enter) to continue:

File Area #   6 ... Acorn Atom
Area change          Locate                Files list          Goodbye
Upload (send)        Download (receive)    Statistics          MAIN MENU
Wild-card search     Raw dir               Copy File > andere Area
DowNload for SysOps Outside
Select: G

Are you sure you want to log-off [y/n] : Y

Leave a note to the Sysop [y/n] ? N
```

[Er verschijnt tenslotte nog een lijst met andere Acorn-BBSsen.]

****

Einde Inlogsessie op UniCorn BBS.
Het is de bedoeling dat er op uniCorn BBS de volgende uitbreidingen gaan plaatsvinden.

* Big Ben Club SoftWare voor de Archimedes (Nu al gedeeltelijk)
* Big Ben Club SoftWare voor de BBC (geplanned)
* Acorn BBC SoftWare Big Ben Club Regio Gooi & Eeemland Geplanned).
* Acorn Atom Club SoftWare (geplanned).
* Atom in PC SoftWare (UpDates) (Nu al aanwezig.)

Uiteraard kunt je ook meedoen aan het EchoMail en NetMail gebeuren. Daarvoor kun je gebruik maken van AcoNet 77:8500/*.* en FidoNet 2:2801/*.* Nog mooier is het als u Point wordt van UniCorn BBS.
Je kan dan bijna alle Nederlandse en ook Engelstalige Acorn message Area's lezen en beschrijven.
Voor meer informatie: schrijf een priveberichtje aan de SysOp.

Ik hoop dat UniCorn BBS ook de 8-bits computerliefhebbers het naar de zin kan maken. Inloggen, of zelfs Point worden is bij UniCorn, op de telefoonkosten na, GRATIS. Voor de Acorn Archimedes is er ook een Pointprogramma WimpLink 1.01 als ShareEare beschikbaar.

Veel succes en welkom in DataLand.

MvGr. |_|
| |enri.

```
10 REM ROTATE
20 REM SCHUIF ALLES VAN RECHTS NAAR LINKS
30 REM BIJDRAGE VOOR INTIKKEN EN RUNNEN MAAR
40 REM LEENDERT 31/10/1994
50 DIM LL20
60 FOR N=1 TO 20; LLN=-1; NEXT
70 FOR N=0 TO 2 STEP 2
80 P=#7000;[
90 :LL0 LDA @#80
100 STA #71
110 LDA @#00
120 STA #70
130 LDX @24
140 :LL3 LDY @#1F
150 CLC
160 :LL4 LDA (#70),Y
170 ROL A
180 PHA
190 DEY
200 BPL LL4
210 LDA @#00
220 STA #72
230 LDY @#00
240 :LL5 PLA
250 STA (#70),Y
260 INY
270 CPY @#1F
280 BNE LL5
290 PLA
300 CLC
310 STA (#70),Y
320 LDA #70
330 CLC
340 ADC @#20
350 STA #70
360 BCC LL3
370 INC #71
380 DEX
390 BNE LL3
400 RTS;]
410 NEXT
420 CLEAR 4
430 MOVE 10,10;DRAW 256, 192
440 DO
445     LINK LL0
446 UNTIL 0
450 END
```

Even een lijntje rechttrekken ...

door roland leurs

In het vorige nummer van Atom Nieuws schreef Leendert het fantastische verhaal over de theorie die schuil gaat achter het tekenen van een lijn. Dat was HET artikel waar ik al ruim een jaar op zat te wachten. Deze interesse wordt opgewekt door de Atom-in-PC (had u anders verwacht?). In versie 3.xx van het terminal programma worden immers de grafische mogelijkheden meer uitgebreid en één van die uitbreidingen is het trekken van een lijn.

Ofschoon we dat al lang kunnen doen met MOVE en DRAW zit aan die werkwijze een enorm groot nadeel. Stel we tekenen de volgende lijn:

```
MOVE 0,0;DRAW 1000,0
```

wat gebeurt er dan? De Atom berekent met zijn eigen ROM-routines welk pixel geplot moet worden en springt dan naar de plotvector om het pixel daadwerkelijk op het scherm te zetten. In een extended video mode wordt dan een commando gegegeven om een pixel op het scherm te zetten, gevolgd door de x en y coördinaten en de plotmode. Even rekenen:

commando:	2 bytes	
x-coördinaat:	2 bytes	
y-coördinaat:	2 bytes	
plotmode:	1 byte	
Totaal:	7 bytes	om één pixel te plotten !

Dat zijn voor bovengenoemde lijn dus 7000 bytes die overgezonden moeten worden om een lijn te tekenen. Dat is nog meer dan het hele SNAPPER programma. En omdat alles lekker via de trage I/O bus van de PC gaat duurt dat lang. Zeker als je 256 van die lijnen wil tekenen.

Daarom heb ik het terminal programma uitgebreid met een commando om een lijn te tekenen. In totaal worden er dan per lijn 10 bytes overgezonden. Op een 486DX 33 MHz is kost het tekenen van een lijn dan nog maar ongeveer 1/3 van de tijd vergeleken met de ouderwetse manier.

Maar goed, de titel van het verhaal geeft aan dat ik een lijntje recht wil zetten en dat is wel nodig. Want mijn routine heb ik afgeleid van de listing van Leendert. Maar helaas maken grote geniën ook fouten.

De tekst over het Bresenham algoritme lijkt mij geheel duidelijk, zeker met de getekende voorbeelden. Maar in het

stroomdiagram op bladzijde 54 zitten twee enge bugs. De eerste staat in het vijfde blokje [van bovenaf geteld]. Daar staat in

Tel Del- taX op by Error- term	dat moet zijn:	Tel Del- taY op by Error- term

In de volgende stap wordt ErrorTerm dan vergeleken met halfX; daar is niets mis mee maar als ErrorTerm > halfX is moet Y aangepast worden. Het stroomdiagram laat in dat geval de sprong uitvoeren om te testen op het laatste pixel.

Verander in die vergelijking dus > in ≤ of wijzig de Y(es) naast de vergelijking in een N(o). Daarmee komen de tekst en het stroomdiagram weer met elkaar overeen waardoor het verhaal makkelijker te begrijpen is.

Dan staan er nog twee bugs in de listing. In regel 570 wordt naar het verkeerde label gesprongen. Er moet daar gesprongen worden naar LL10 om het volgende pixel te plotten. (Leendert laat de routine naar LL11 springen waardoor steeds ErrorTerm op de beginwaarde gezet wordt. Uit ervaring weet ik dat sommige lijnen wat veel correcties krijgen en dan heel ergens anders eindigen dan gepland.)
De tweede fout is wat duidelijker te zien en staat in regel 700. Daar staat de sprong naar het einde van de routine; helaas springt-ie één byte te kort en gaat dan nog wat tekenen.

Wijzig dus in de listing de volgende twee regels in:

```
570 JMP LL10          \ plot de coördinaten
700 BEQ LL9           \ indien gelijk dan einde routine
```

Dan zit er nog een doordenkertje in de listing, zeker als je hem vergelijkt met het stroomdiagram. In het diagram wordt ErrorTerm vergeleken met halfX; in de listing (zie regels 720 en 730) wordt getest of ErrorTerm positief of negatief is. Dit lijkt heel verschillend maar in het programma wordt ErrorTerm in het begin van de routine (regels 590 t/m 635) niet gelijk gemaakt aan 0 (zoals in het stroomdiagram staat) maar aan halfX. Daardoor kan de vergelijking later met 0 gemaakt worden. Vandaar de test op een positieve of negatieve ErrorTerm.

Hopelijk is het een en ander nu wat duidelijker,
Leendert nogmaals bedankt voor de uitleg.

Met vriendelijke groeten,
Roland Leurs

Een gestructureerde manier van ontwerpen

Inleiding

Het schrijven van een programma is een ware kunst. Toch is zoiets niet alleen een kwestie van Jobs geduld en Salomo's wijsheid. Er zijn diverse methoden om het schrijven van een programma in goede banen te leiden.
In dit stuk wordt een methode beschreven voor het opzetten van de structuur voor een programma.
Zoals voor alle dingen in het leven: deze methode leer je door het te doen en te gebruiken. Natuurlijk zal het zo zijn dat het lastig lijkt op het eerste moment, natuurlijk zal het zo zijn dat het allemaal heel anders kan. Maar het is een methode en pak er uit op wat je kunt gebruiken.

De methode

Iedereen werkt volgens een bepaalde methode bij het schrijven van een programma. De meest populaire noem ik u: JBF, de Jan Boeren Fluitjes methode. Niets plannen. Nee, gewoon een avondje gaan zitten en kloppen maar.
Een andere iets minder bekende maar zeker net zo vaak toegepast is methode Heijermans. De naam van deze methode is afgeleid van een titel van een van de boeken van H. Heijermans: op hoop van zegen. Ook niet echt iets waar de sfeer van professionaliteit van afdruipt dus.
Verder nog wat professionele methoden Yourdon, als variant op de beroemde tandeborstel. De analogie gaat verder: net als die tandeborstel moet je ook de methode om de drie maanden vervangen.
Nasi Scheiderman, als variant op de Indische keuken. Erg handig, maar vaak wat chaotisch als het op de details aankomt.
De methode die ik hier wil gaan behandelen is de JSP, de Jackson Structured Programming methode. Door boze tongen ook wel vertaald met Jackson Slow Programming. Maar oordeelt u zelf.

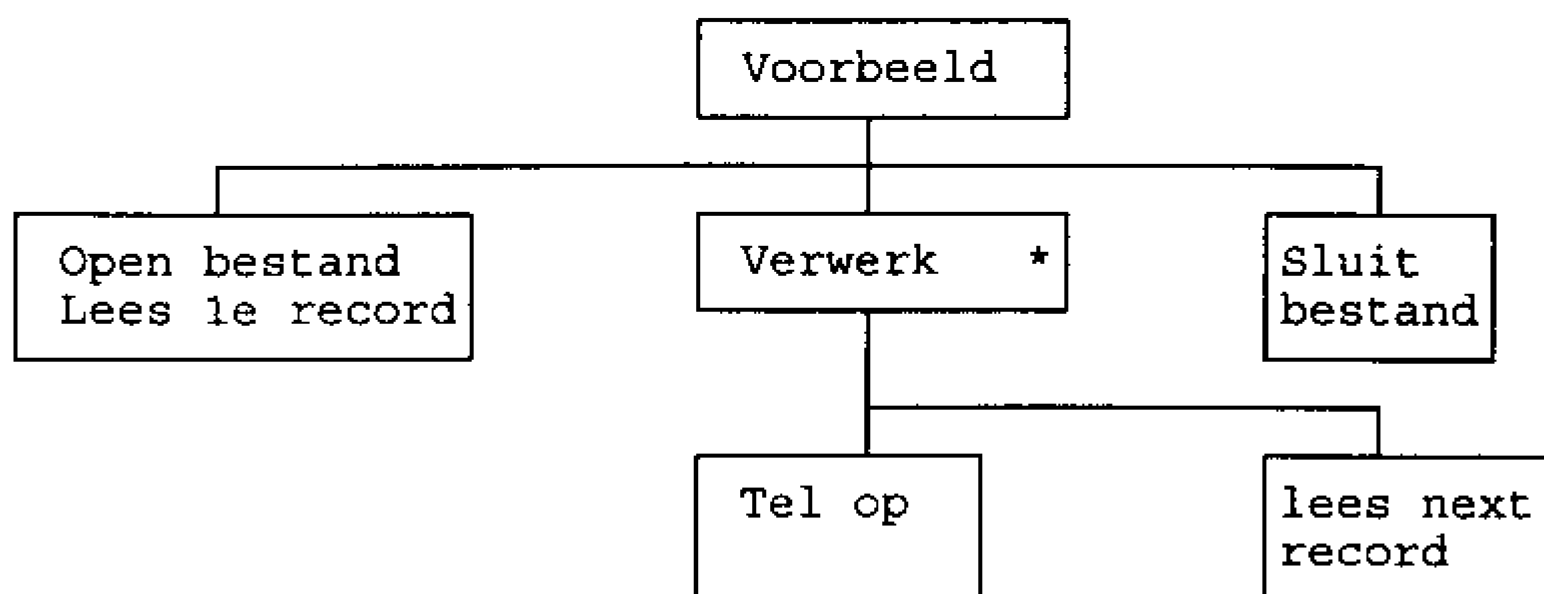
JSP

De genoemde methode is bedacht door een Amerikaanse wiskundige genaam Michael Jackson, niet te verwarren met onze zingende

kindervriend. Deze methode is erop gebasseerd dat alle zaken een begin conditie hebben, een iteratie (een do-until) en een eind conditie. Klinkt allemaal wat verheven maar werkt erg logisch. Kijkt u maar even mee in het volgende voorbeeld:

```
1         Open bestand
          Lees 1e record
2         while not (EOF)
            verwerk record
            lees record
          end while
3         sluit bestand
```

Het aardige van de methode is, dat je hem volledig uit kan tekenen. Iedere situatie beschijft de functionaliteit in ene blokje. Zo valt bovenstaand voorbeeld in onderstaand plaatje te vatten



In de methode zijn wat afspraken gemaakt over notaties. Ieder blokje beschijft een functie, zo u wil subroutine. Het sterretje in verwerk duidt een iteratie aan. Meestal wordt de contitie erbij geschreven. In dit geval zou dus bij verwerk de conditie 'until EOF' worden geschreven.
Staat op de plaats van het sterretje een rondje, dan wordt een IF conditie aangegeven. Ook hier geldt, dat de conditie wordt opgeschreven.
Doordat in de blokken de naam van de subroutine wordt aangegeven, automatisch een onderscheid gemaakt tussen statements en het aanroepen van subroutines. Goto's komen in dit plaatje helemaal niet voor. Hiermee wordt weer eens aangetoond dat 'goto' less programmeren geen doel op zich is, maar dat de gekozen methode het gebruik van goto's helemaal overbodig maakt.

Bovenstaand JSP diagram, want zo heet zo'n ding officieel, kan worden vertaal in onderstaand programma:

```
 10 PROGRAM voorbeeld
 20 REM JSP demo
 30 PROC_Init

 40 WHILE NOT F_EOF
 50   PROC_Verwerk
 60 WEND
 70 PROC_Close
 80 END
 90 REM
100 DefProc_Init
105 F_EOF=FALSE;Totaal=0
110 a%=openin"bestand"
120 e$=fget$(#a%)
130 if EOF(#a%) then F_EOF=TRUE
135 ENDPROC
137 REM
140 DefProc_Verwerk
150 Totaal=Totaal+ (gegevens uit record)
160 e$=fget$(#a%)
170 if EOF(#a%) then F_EOF=TRUE
180 ENDPROC
190 rem
200 DefProc_Close
210 fclose(#a%)
220 ENDPROC
```

Het aardige is dat er een duidelijke relatie ligt tussen het plaatje en het programma. Sterker nog: Als een van beide ontbreekt, is het toch mogelijk het andere te reproduceren. In de praktijk is het zo dat programmeurs die vanuit het zelfde JSP diagram werken, tot de zelfde oplossing komen.

En nou verder

Leuk, aardig en lastig. Dat is wellicht het gevoel wat u nu heeft. Maar aan de hand van een aantal voorbeelden wil ik u meevoeren naar de mogelijkheden van deze methode. Hebt u de methode eenmaal onder de knie, dan zijn er geen programma's meer die te groot of te complex zijn. Immers alle stukken vallen op te delen in eenvoudige brokken.

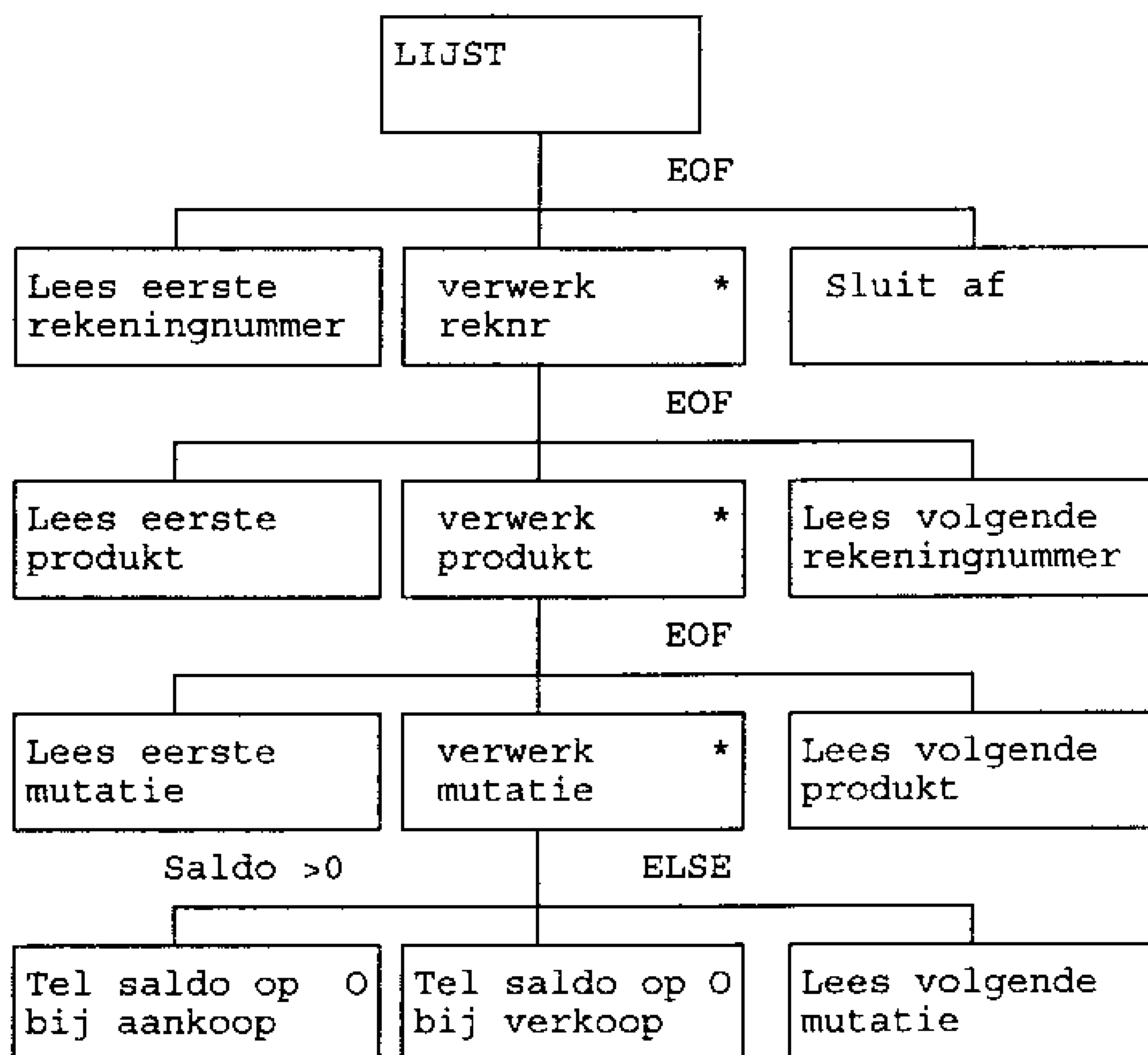
Een voorbeeld

Tegenwoordig heet alles een Case. Dus waarom ook wij niet onze eigen case.
We maken een lijst van rekeningnummers. Ieder rekeningnummer komt op een apart blad. Ieder rekeningnummer heeft echter verschillende componenten, te weten aandelen, obligaties, futures, opties en warrants. Achter ieder component kunnen 0 of meer mutaties zitten. U koopt aandelen, dat resulteert dan in een toename van het aantal aandelen. Verkoopt u ze, dan, jawel, neemt het aantal aandelen af. Zo ook voor de overige componenten. Het totaal van de aan en verkopen wordt op het lijstje vermeld, zodat onderstaand lijstje verschijnt:

Rekeningnummer: xx.xx.xxx.xxx

	aankoop	verkoop
Aandelen:	x.xxx,xx	
Obligaties:	..	
Futures:	..	
Opties:	..	
Warrants:	..	

Na dit inleidend verhaal is het al mogelijk een JSP diagram te maken van bovengenoemde analyse. Daar gaan we:



Uitgewerkt in coding levert dat qua principe het volgende programma op. Gemakshalve gebruikt ik wat leesbare statements die om wille van de duidelijkheid niet geheel Basic zijn.

```
10 PROGRAM LIJST
20 PROC_INIT_REKENINGNUMMER
30 DO
40    PROC_VERWERK_REKENINGNUMMER
50 UNTIL REK_EOF
60 PROC_ENDING
70 END

100 DEF PROC_INIT_REKENINGNUMMER
110 OPEN "REKENINGNUMMER" FOR INPUT AS #1
120 REK_EOF = READREC REKENINGNUMMER, INVOERRECORD
199 ENDPROC

200 DEF PROC_VERWERK_REKENINGNUMMER
210 PROC INIT_PRODUKT
220 DO
```

```
230   PROC_VERWERK_PRODUKT
240 UNTIL PRODUKT_EOF
250 PROC_END_PRODUKT
299 ENDPROC

300 DEF PROC_INIT_PRODUKT
310 OPEN "PRODUKT" FOR INPUT AS #2
320 PRODUKT_EOF = READREC PRODUKT, INVOERPRODUKT
399 ENDPROC

400 DEF PROC_VERWERK_PRODUKT
410 PROC_INIT_MUTATIES
420 DO
430   IF MUTATIESALDO > 0
440      PROC_TEL_MUTATIE_OP
450   ELSE
460      PROC_TEL_MUTATIE_AF
470   UNTIL MUTATIE_EOF
480 PROC_READ_NEXT_MUTATIE
499 END

500 DEF PROC_INIT_MUTATIES
510 OPEN "MUTATIES" FOR INPUT AS #3
520 MUTATIES_EOF = READREC MUTATIES, INVOERMUTATIE
599 ENDPROC

600 DEF PROC_TEL_MUTATIE_OP
610 LIJSTJE[PRODUKT,1]+SALDO
699 ENDPROC

700 DEF PROC_TEL_MUTATIE_AF
710 LIJSTJE[PRODUKT,2]-SALDO
799 ENDPROC

800 DEF PROC_READ_NEXT_MUTIE
810 MUTATIE_EOF = READREC MUTATIE, INVOERMUTATIE
899 ENDPROC

900 DEF PROC_READ_NEXT_PRODUKT
910 PRODUKT_EOF = READREC PRODUKT, INVOERPRODUKT
999 ENDPROC

1100 DEF PROC_END_PRODUKT
1110 CLOSE MUTATIES
1199 ENDPROC
```

```
1200 DEF PROC_READ_NEXT_REKENINGNR
1210 CLOSE PRODUKT
1220 PRINT GENOEMD LIJSTJE
1230 REKNR_EOF = READREC REKENINGNR, INVOERREKENING
1299 ENDPROC

1300 DEF PROC_ENDING
1310 CLOSE REKENINGNR
1399 ENDPROC
```

Al met al best omvangrijk qua omvang, maar qua toegankelijkheid moet het toch ook wel iets zijn. Zoals gezegd: tik het programma niet over wat het zit vol met niet-basic opdrachten. Ik verdenk u ervan dat u de vertaalslag naar de juiste syntax zelf kunt maken.

Probeert u zelf eens een klok te maken op basis van een JSP methode. Denk daarbij aan het feit dat er 60 seconde in 1 minuut gaan; 60 minuten in 1 uur en 24 uren in 1 dag. Gebruik de variabelen S, M en U. Doe de iteratie met de genoemde getallen met een DO-UNTIL of iets dergelijks. Maak eerst een JSP diagram en schrijf daarna de code. Stuur het resultaat eens op aan de redactie en ik meld u een van de volgende keren of er überhaupt reactie is gekomen, want de Atomisten zijn blijkbaar meer een lezend dan een reagerend volkje geworden, en wat er eventueel valt te verfijnen aan de methode of de uitleg ervan.

Conclusie

De JSP methode kan vele voordelen bieden. Het belangrijkste voordeel is dat het een methode is die taal onafhankelijk is en bovendien in een afbeelding kan worden gegoten. Door de opzet van de methode wordt ook slechts op eèn plaats in de code een bepaalde functie gedaan. Gaat in het genoemde voorbeeld het lezen van een mutatie niet goed, dan heeft het geen zin om in de procedure die het rekeningnummer leest te kijken, nee, de procedure waar de mutaties worden gelezen behoeft dan nader onderzoek. Zeker in grote programma's is dat toch wel ideaal.

Atom-in-PC: Meest gestelde vragen.

door roland leurs

Regelmatig krijg ik telefoontjes van mensen die enkele vragen hebben met betrekking tot de inmiddels zeer veel gekochte Atom-in-PC kaart. In dit artikel zal ik deze vragen nogmaals beantwoorden. Niet omdat ik geen telefoon wil, integendeel zelfs, maar zodat u sneller uw vragen beantwoord krijgt.

Vraag: Moet er per se een R65C02 processor gebruikt worden?
Antwoord: Nee, u kunt gerust een 6502 processor gebruiken, op 1 of 2 MHz. De snelheid kan ingesteld worden met de jumper bij het klokcircuit. Zelfs de 65802 kan gebruikt worden.

Vraag: Na het opstarten van het terminal programma blijft het scherm zwart of er verschijnen doorlopend karakters op het scherm. Waar ligt dat aan?
Antwoord: In de meeste gevallen vindt de PC in het adres-gebied de kaart niet. Omdat de controle nogal erg eenvoudig is kan het voorkomen dat de PC "iets" ziet op de controle-adressen en het domme ding "denkt" dan dat de kaart aanwezig is op dat adres. De oplossing is dan het meegeven van de /A=aaa optie. Als bijvoorbeeld de kaart geadresseerd is op I/O adres 310h geef dan de volgende optie mee:

```
ATOM /A=310
```

Vraag: Waarom laten enkele *-commando's laten het hele systeem vastlopen?
Antwoord: Door een foutje in de versies t/m 2.15 van het terminalprogramma vindt de software de naam van de DOS commando interpreter niet als deze niet als eerste in de DOS environment staat. Vanaf versie 2.20 is deze fout verholpen.

Vraag: Waar kan ik de meest recente software krijgen?
Antwoord: Op het Unicorn BBS van Henri Derksen; telefoonnr 085-425506.

Deze lijst zal regelmatig aangevuld worden. U kunt deze dan downloaden van bovenstaand BBS. Mocht u nog vragen hebben, bel of schrijf gerust. Sinds kort kan het zelfs per E-Mail:

Roland Leurs
Prins Mauritslaan 43
6191 EC Beek

telefoon: 046-370650
fidonet: 2:285/226.9

REGIO-ADRESSEN.

Wilt U lid worden van de ATOM COMPUTER CLUB ?.
Neem dan contact op met de penningmeester van de regio waar U bij ingedeeld wenst te worden. Deze kan U inlichten omtrent het lidmaatschap.

Regio NOORD-HOLLAND :
P.v.Kuik, Zuideinde 54-a, 1843 JP Groot Schermer.
tel. 02997- 1902.

Regio DEN HAAG +ARNHEM
Th.Waayer, H.v.Boeijenlaan 66, 2273 DC Voorburg.
tel. 070-3862504.

Regio BRABANT-OOST + ZEELAND
J.Teulings, K.Doormansstraat 54, 5224 GL Den Bosch.
tel. 073-212888.

Regio LIMBURG + BELGIE EN OOST/NOORD NEDERLAND
C.Rutkowski, Mgr.Buckstr.8 6121 KV Urmond
tel. 04498- 60136

Leden die behoren tot opgeheven regio's, danwel regio's die conform de statuten geen lid meer zijn van de federatie, worden in verband met de financ. administratie en de verzending van ATOM-NIEUWS,door de federatie toegewezen aan de nabije regio's.
Zo men tegen deze indeling bezwaar heeft, om welke reden dan ook, kan men de regio van eigen keuze opgeven aan de penningmeester van de federatie: T.Rutten , zie de pagina van de federatie in dit blad.

Bij het aangaan van het lidmaatschap kunt U de contributie overmaken op de rekening van de federatie. Vermeld hierbij uw volledige naam, en adres , alsmede evt. de regio waarbij U wenst te worden ingedeeld.